HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

DIPLOMARBEIT

Herr
Jens Böttcher

Statistische Analyse großer Datenmengen mittels Oracle Data Mining 11g

Mittweida, 2011

Fakultät Mathematik
/Naturwissenschaften/Informatik

DIPLOMARBEIT

Statistische Analyse großer Datenmengen mittels Oracle Data Mining 11g

Autor:
Herr Jens Böttcher

Studiengang:
Angewandte Mathematik

Seminargruppe:
MA05w1

Erstprüfer:
Prof. Dr. rer. nat. Egbert Lindner

Zweitprüfer:
Dipl.-Wirtsch.-Inf. Marco Fischer

Einreichung:
Mittweida, 01.08.2011

Verteidigung/Bewertung:
Mittweida, 2011

Bibliografische Beschreibung:

Böttcher, Jens:
Statistische Analyse großer Datenmengen mittels Oracle Data Mining 11g. - 2011. -100 S. Mittweida, Hochschule Mittweida, Fachbereich Mathematik, Diplomarbeit, 2011

Referat:

Diese Diplomarbeit befasst sich mit der Untersuchung der Data Mining Verfahren der Zusatzoption „Data Mining“ der Oracle Datenbank 11g. Es werden die Verfahren des Data Mining beschrieben. Im Rahmen einer Teststudie wird dargestellt, wie man einige Fragestellungen mit Hilfe des Oracle Data Miners lösen kann.

Inhalt

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

CART ..... Classification and Regression Trees
CHAID .....Chi-square Automatic Interaction Detectors
CMI .....Case Mix Index
CRISP-DM .....Cross-Industry Standard Process for Data-Mining
FAB .....Fachabteilung
ICD International Statistical Classification of Diseases and Related Health Problems
ID3 .....Iterative Dichotomiser 3
IK..... Institutskennzeichen
InEK .....Institut für das Entgeltsystem im Krankenhaus
KDD ..... Knowledge Discovery in Databases
KHH .....Krankenhaushäufigkeit
OLAP..... On Line Analytical Processing
OPS..... Operationen- und Prozedurenschlüssel
SMS.....Sächsisches Staatsministeriums für ..... Soziales und Verbraucherschutz
SMS-ZFD .....Sächsisches Staatsministeriums für .....Soziales und Verbraucherschutz - .....Zentrale Fachdatenbank
SQL .....Structured Query Language
SVM .....Support Vector Machine

1 Einleitung

1.1 Motivation

Durch die immer größere werdenden Datenmengen ist es nötig Data Mining Algorithmen und Verfahren auf große Datenmengen anzuwenden. Oracle bietet eine „Data Mining“ Option für die Oracle Datenbank 11 g an. Dafür werden eine Vielzahl von Algorithmen, statistischen Verfahren und Problemtypen angeboten, um Aufgabestellungen lösen zu können. Von vielen Unternehmen werden Data Mining Systeme für kunden- und marketingbasierte Zwecke benutzt um somit einen Vorteil gegenüber der Konkurrenz zu haben. Diese Möglichkeiten allein sind Grund genug um sich einmal näher mit diesen Systemen zu beschäftigen.

1.2 Zielsetzung

In dieser Arbeit wird für die Robotron Datenbank Software GmbH untersucht inwiefern man die Zusatzoption „Data Mining“ der Oracle Datenbank 11g für eine sinnvolle Analyse von Problemstellungen benutzen kann.

Am Beispiel der Abrechnungsdaten von Behandlungen in sächsischen Krankenhäusern, ggf. auch von weiteren Anwendungsdaten, sollen die Möglichkeiten der automatisierten Identifikation von Mustern und Zusammenhängen in diesen Daten und die daraus möglichen Ableitungen von Prognosen untersucht werden.

Es soll eine detaillierte Bewertung der in der Oracle Data Mining Option zum Einsatz kommenden Algorithmen und statistischen Verfahren er-

folgen, mit einer Zuordnung, für welche Art von Problem- bzw. Aufgabenstellung diese geeignet sind.

1.3 Aufbau der Arbeit

Die Grundlagen des Data Mining / Knowledge Discovery in Databases werden im Kapitel 1 dargelegt. Dabei wird Data Mining vorerst definiert. Anschließend wird ein Ablaufplan für das Data Mining betrachtet. Folgend werden die möglichen Datentypen, die bei einer Data Mining Analyse auftreten können, beschrieben. Eine Auswahl der möglichen Datenaufbereitungsmöglichkeiten für die Data Mining Analyse wird danach beschrieben. Weiterhin werden die verschiedenen Problemtypen des Data Mining beschrieben. Anschließend werden für diese Problemtypen einige Algorithmen vorgestellt, die für die Lösung benutzt werden können. Einige Verifizierungsmethoden werden dargestellt und es werden Möglichkeiten zur Verbesserung von Klassifikatoren untersucht. Weiter wird das Oracle Data Mining und speziell der Oracle Data Miner, eine grafische Benutzeroberfläche, vorgestellt. Zum Schluss wird in einer Fallstudie überprüft, inwiefern Fragestellungen mit den Data Mining Algorithmen und dem bereits beschriebenen Ablaufplan gelöst werden können.

2 Data Mining – Grundlagen

In diesem Kapitel werden zuerst die Begriffe „Knowledge Discovery in Databases“ und „Data Mining“ begrifflich abgegrenzt. Es wird besonders ein Ablaufplan für das Data Mining betrachtet. Im Laufe des Kapitels werden einige gängige Problemtypen für das Data Mining erläutert. Dazu werden die zu den Problemtypen gehörigen Algorithmen in Auszügen vorgestellt.

2.1 Einführung - Data Mining

„Schätzungen besagen, dass sich die Menge der in allen Datenbanken der Welt abgelegten Daten alle 20 Monate verdoppelt. “ (Witten & Frank, 2001) *S. 2* Diese Einschätzung aus dem Jahr 2001 wurde jedoch bereits 2004 durch: *„The (…) exponential growth of data, estimated by Greg’s Law (…) double[s] on average every nine months. “* (Netezza Corporation, 2004) korrigiert.

Durch immer größer werdende Datenmengen genügt es nicht mehr, in den Daten mithilfe von Methoden der Statistik und / oder manuell nach Mustern, die z. B. für ein Unternehmen oder eine Behörde wichtig sein könnten, zu suchen. Aus diesem Grunde mussten neue Analysemethoden gefunden werden, die diese Hürden in einer akzeptablen Zeit meistern können. Hier setzt das Data Mining bzw. „Knowledge Discovery in Databases“ (KDD) an.

Im weiteren Verlauf soll zuerst, um das Verständnis für das Data Mining zu festigen, auf die Grundlagen und eine Definition des Data Mining eingegangen werden. Um Data Mining erfolgreich anwenden zu können,

muss zunächst geklärt werden, wie Data Mining in der Praxis definiert wird und welche Schritte im Data Mining Prozess nötig sind. Ebenso ist es wichtig zu wissen, welche Algorithmen oder Verfahren aus den unterschiedlichen Fachgebieten eingesetzt werden können.

Data Mining ist mit einer Menge von Fachgebieten verwandt. Dazu zählen u. a. das Maschinelle Lernen, die Statistik und die Optimierung. Viele Verfahren aus diesen Fachgebieten können ebenfalls zum Data Mining benutzt werden.

Das wesentliche Ziel des Data Mining ist es Muster zu finden. Data Mining wird daher in der Fachliteratur auch als Mustererkennung bezeichnet. vgl. (Fayyad, Piatetsky-Shapiro, Smyth, & Uthurusamy, 1996) S. 4, (Witten & Frank, 2001) S: 2 Laut Han bezieht sich „Data Mining“ auf die Herausfilterung von Wissen aus großen Datenmengen. vgl. (Han & Kamber, 2001) S. 5

Data Mining wird in der Literatur auch als „Knowledge Discovery in Databases“ bezeichnet. Dadurch wird das eigentliche Ziel von Data Mining, die automatische Extraktion von Mustern und Wissen aus großen Datenbeständen, stärker heraus gearbeitet. Dieser Begriff wurde von Usama Fayyad erstmals eingeführt und folgendermaßen definiert:

„Knowledge discovery in databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. “ (Fayyad et al., 1996) S. 6

Es ist also der nicht-triviale Prozess um gültige, neuartige, potentiell nützliche, und letztendlich verständliche Muster in Daten zu identifizieren.

Diese Definition wird im weiteren Verlauf detaillierter erläutert.

Sei F die Menge aller Datensätze, d.h. alle Fälle die den Sachverhalt wiedergeben. Ein Muster ist ein Ausdruck E in einer Sprache L, der Datensätze in $F_E \subseteq F$ beschreibt. E wird als Muster bezeichnet, falls es einfacher ist als die Auflistung aller Datensätze in F_E. Nicht trivial bedeutet in diesem Zusammenhang, dass die gefundenen Muster nicht durch einfache Abfragesprachen wie z. B. SQL, OLAP oder durch einfache statistische Kenngrößen wie dem Mittelwert oder durch Aggregieren einiger Werte zu erhalten sind. Die Muster sollten gültig im Hinblick auf die Anwendbarkeit der Muster auf neue Daten sein, da die Muster sonst nur schwer zu verallgemeinern wären. Dabei kann man ein Gewissheitsmaß C definieren, das Ausdrücke der Sprache L in einen teil- oder totalgeordneten Maßraum M_C abbildet. Ein Ausdruck E in L über $F_E \subseteq F$ kann einem Gewissheitsmaß $c = C(E,F)$ zugeordnet werden. Ebenso ist es verständlich, dass die Muster in einem gewissen Maße neuartig sein sollen, da schon bekannte Muster keinen Mehrwert für ein Unternehmen oder eine Behörde bedeuten. Diese Muster sollten natürlich auch einen potentiellen Nutzen für ein Unternehmen oder eine Behörde darstellen, wie z. B. die Erhöhung des Umsatzes für ein bestimmtes Produkt oder eine bestimmte Produktgruppe. Oft ist es aber gewünscht, dass die Muster für einen Menschen verständliche Muster erzeugen z. B. um mit diesem Wissen einen besseren Einblick und Verständnis für die Problemstellung zu gewinnen. Analog zum Gewissheitsmaß kann man auch Maße für diese Eigenschaften finden. Oft wird auch von einem sogenannten Interessantheitsmaß eines Musters, in diesem Zusammenhang, gesprochen. Dieses vereint die Maße der anderen gewünschten Eigenschaften eines Musters. Mit Hilfe dieses Interessantheitsmaßes kann man Muster auch nach dem Wissensgehalt vergleichen und somit poten-

tiell neues Wissen anhand von Schranken für das Interessantheitsmaß bestimmen. vgl. (Fayyad et al., 1996) S. 7

Der KDD-Prozess beinhaltete das Data Mining ursprünglich nur als einen Teilprozess. Fayyad definierte Data Mining folgendermaßen:

„Data Mining is a step in the KDD process consisting of particular data mining algorithms that, under some acceptable computational efficiency limitations, produces a particular enumeration of patterns E_j *over* F*. „ (Fayyad et al., 1996) S. 9*

Mittlerweile werden beide Begriffe aber als Synonym für den ganzen Data Mining Prozess gebraucht. Im weiteren Verlauf dieser Arbeit wird der Begriff Data Mining für den ganzen KDD-Prozess verwendet.

2.2 CRISP-DM

Um ein Data Mining Projekt erfolgreich durchführen zu können ist es notwendig einen geeigneten Ablaufplan für das Data Mining zu finden, so dass der Prozess für viele Problemstellungen sinnvolle und anwendbare Ergebnisse liefert und jederzeit reflektiert werden kann. Der Cross-Industry Standard Process for Data-Mining (CRISP-DM) ist ein solcher Ablaufplan. Der CRISP-DM Prozess wurde von einem Konsortium namhafter Unternehmen, unter anderem SPSS und Daimler Chrysler, entworfen. Dieser ist dem KDD-Prozess in den Grundzügen sehr ähnlich. Aus diesem Grunde wird in dieser Arbeit nur der CRISP-DM Prozess näher vorgestellt.

Die einzelnen Schritte des CRISP-DM werden im Folgenden in Anlehnung an (Chapman et al., 2000) näher beschrieben.

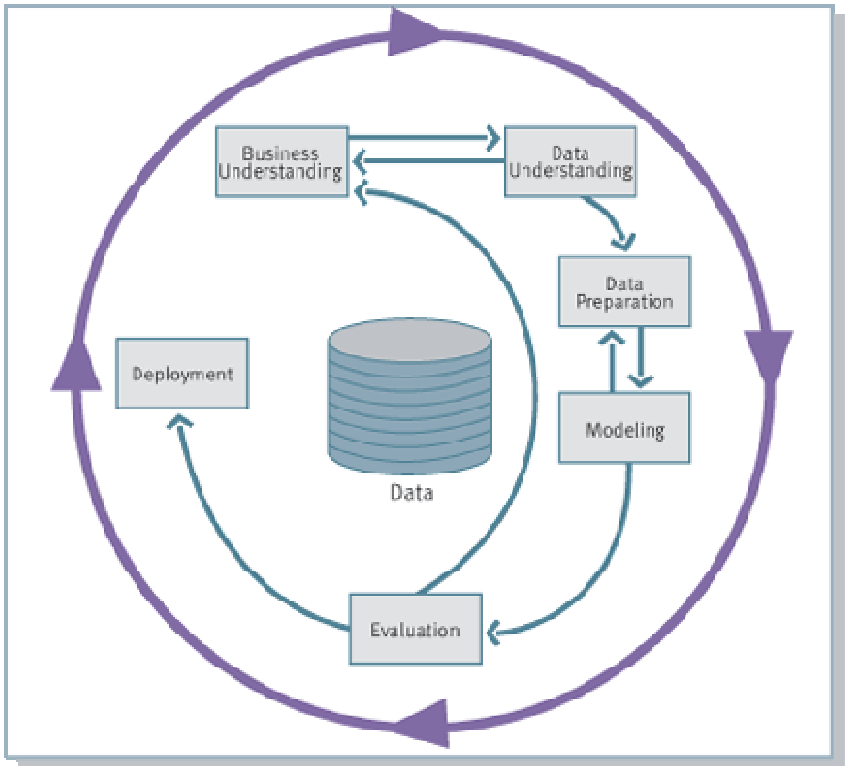

Abbildung 1: CRISP-DM (Chapman et al., 2000)

In Abbildung 1 wird der Ablauf von CRISP-DM dargestellt.

Der erste Schritt im Ablaufplan ist das „**Business Understanding**“ und beinhaltet, dass man die Ziele und Erfordernisse des Projektes aus einer geschäftlichen Sicht versteht. Dieses Wissen wird anschließend in eine Data Mining Problemstellung überführt und es wird ein vorläufiger Ablaufplan erstellt. Ohne zu wissen, nach was gesucht werden könnte kann ein Data Mining Projekt nicht sinnvoll angegangen werden.

Unter „**Data Understanding**“, im zweiten Schritt, versteht man, dass man die Daten vorher verstehen muss, um adäquat mit ihnen umgehen zu können. Dazu gehört, dass man eine erste Auswahl an Daten vornimmt, mit denen man das Data Mining durchführen will. Weiterhin beinhaltet dieser Schritt, dass man weiß welche Daten vorhanden sind und was diese aussagen. Ebenso ist es wichtig mittels geeigneter Tests herauszufinden inwiefern diese Daten für eine Analyse verwendet werden

könnten. Dieser Schritt beinhaltet außerdem, dass man interessante Teilmengen der Daten identifiziert und Hypothesen über versteckte Informationen aufstellt. Dabei kann man die Daten mittels verschiedener Verfahren wie Histogramme, Plots, … visualisieren. In diesem Schritt sollte man aber auch mögliche Fehler in den Daten finden. Diese könnten die Ergebnisse des Data Mining unter Umständen stark beeinflussen.

Beim Punkt „**Data Preparation**“ werden alle Verfahren und Möglichkeiten zusammengefasst, wie man die Daten richtig vorbereiten kann, um möglichst gute Ergebnisse beim Anwenden der Data Mining Methoden zu erhalten. Dabei wird auf die Erkenntnisse aus dem „Data Understanding“ zurückgegriffen, um die Daten so besser vorbereiten zu können. Zu den Methoden zählen u. a. verschiedene Transformationen, das Bereinigen der Daten und die Auswahl von wichtigen Attributen.

Der vierte Schritt im Ablaufplan, das „**Modeling**“, umfasst das Ausführen aller möglichen erfolgsversprechenden Problemtypen und Verfahren die für die Behandlung der Data-Mining Aufgabe nötig sind. Dabei werden die vorhandenen Parameter, die für die Verfahren zur Verfügung stehen so lange verändert bis keine Verbesserung mehr möglich ist. Dabei muss man, falls es ein Algorithmus erfordert, notfalls einen Schritt zurück zur „Data Preparation“ gehen.

Die „**Evaluation**“ beinhaltet die Überprüfung, der im Schritt „Modeling“ erzielten Ergebnisse, im Hinblick auf die Erfüllung der Aufgabenstellung. Dabei kann man, falls keine zufriedenstellenden Ergebnisse gefunden wurden, die vorherigen Schritte noch einmal unter veränderten Voraussetzungen oder mit einem anderen Modell wiederholen. Am Ende der Evaluation sollte entschieden werden wie erzielten Ergebnisse weiterverwendet werden sollten.

Falls man, mit Hilfe eines Ablaufplans, ein geeignetes Modell gefunden hat und dieses als zufriedenstellend eingestuft hat, ist es schließlich noch erforderlich dieses Wissen in eine geeignete Form zu bringen und geeignet zu präsentieren. Das wird beim letzten Schritt dem „**Deployment**“ durchgeführt. Dabei ist es oft ausreichend, die Regeln in einer verständlichen Form ausgeben zu lassen. Oftmals ist es auch von Interesse die gefundenen Regeln auf neue Daten anzuwenden.

Die Darstellung des Ablaufplans in Abbildung 1 suggeriert, dass der Plan zunächst bis zur Evaluation vollständig durchlaufen wird. Aus Gründen der Übersichtlichkeit werden einige Rückwärtsschritte in der Grafik weggelassen. Es ist z. B. auch zwischen der „Data Preparation“ und dem „Data Understanding“ ein Zurückspringen vorgesehen. Das ist z. B. dann erforderlich, wenn man während der Analyse der Daten einige Informationen aufdeckt, die sich nicht mit dem bisherigen Verständnis der Daten decken.

Im weiteren Verlauf der Arbeit werden einige der Prozessschritte näher betrachtet werden und einige Verfahren, die sich in der Praxis durchgesetzt haben, detailliert betrachtet. Dabei wird in dieser Arbeit ein besonderes Augenmerk auf die „Data Preparation“ (Datenaufbereitung) und das „Modeling“ (Modellbildung) gelegt werden.

2.3 Datentypen

Für das Data Mining ist es unbedingt nötig, dass man zuerst die möglichen Datentypen versteht, die es beim Data Mining oder bei allgemeineren statistischen Anwendungen gibt. Das ist nötig, da einige Methoden der Datenaufbereitung, Problemtypen und Algorithmen nur mit bestimmten Datentypen funktionieren.

Hier werden die Datentypen in Anlehnung an (Bamberg & Bauer, 1998) klassifiziert.

Die verschiedenen Datentypen muss man formal zwischen qualitativen und quantitativen Datentypen unterscheiden. Bei quantitativen Datentypen werden hauptsächlich Zahlen zur Unterscheidung verwendet. Im Gegensatz dazu werden qualitative Datentypen häufig durch Wörter charakterisiert. Man sollte allerdings beachten, dass qualitative Datentypen ebenfalls durch Zahlen charakterisiert werden können. Aus diesem Grunde kann man die Datentypen weiterhin mit Hilfe von gewissen Skalierungen charakterisieren. Es sind folgende Skalierungen von Datentypen bekannt:

- Nominalskala
- Ordinalskala / Rangskala
- Kardinale Attribute:
    - Intervallskala
    - Verhältnisskala / Ratioskala
    - Absolutskala

Bei der Nominalskala werden die Ausprägungen der Attribute nur untereinander unterschieden ohne dass es eine Ordnung gibt. Als Beispiel für ein nominales Attribut kann man das Geschlecht nehmen, da man nur entscheiden kann ob ein Mensch männlich / weiblich ist oder nicht. Man kann aber keine Ordnung für das Geschlecht definieren. Die Nominalskala beschreibt aus diesem Grunde eine Äquivalenzrelation zwischen den Attributen.

Eine Äquivalenzrelation ist eine Relation die

- reflexiv,

- transitiv und
- symmetrisch

ist.

Die Ordinalskala / Rangskala kennt im Gegensatz zur Nominalskala eine Ordnung unter den verschiedenen Attributwerten. Die Ordinalskala beschreibt dementsprechend eine Ordnungsrelation. Als Beispiel kann man verschiedene Bewertungsskalen von „gefällt mir sehr“ bis „gefällt mir überhaupt nicht“ oder Schulnoten angeben.

Eine Ordnungsrelation ist eine Relation, die die Eigenschaften

- reflexiv,
- transitiv und
- antisymmetrisch

aufweist.

Kardinale Attribute (auch quantitative/numerische Attribute) besitzen ebenso wie die Ordinalskala eine Ordnungsrelation. Im Gegensatz dazu kann man bei kardinalen Attributen aber feststellen, ob ein Wert im Verhältnis mehr aussagt wie ein anderer. Bei der Intervallskala kann man dabei nur die Abstände / Intervalle für das Attribut miteinander vergleichen. Ein Beispiel für eine Intervallskala ist die Temperatur in Celsius. Dafür kann man z. B. nur aussagen, dass es heute 10°C wärmer ist als gestern. Mit Hilfe der Verhältnisskala / Ratioskala, wobei ein Nullpunkt für das Attribut vorhanden ist, kann man darüber hinaus noch aussagen, ob ein Wert ein Vielfaches eines anderen Wertes aussagt. Als Beispiel für eine Verhältnisskala kann man sich die Preise eines Produktes vorstellen. Man kann z. B. aussagen, dass sich ein Produkt in diesem Jahr um

5% verteuerte. Für eine Absolutskala ist zusätzlich noch eine Einheit, wie Stück, definiert.

2.4 Methoden der Datenaufbereitung

In diesem Unterkapitel werden einige Methoden zur Datenaufbereitung und mögl. Probleme der Daten vorgestellt. Die Methoden der Datenaufbereitung werden anhand (Han & Kamber, 2001) dargestellt.

Um Daten überhaupt für das Data Mining benutzen zu können, müssen sie vorher aufbereitet werden. Dafür gibt es verschiedene Methoden.

Viele reale Datensätze besitzen fehlerhafte Daten. Dabei können u. a. folgende Fehler auftreten:

- fehlende Daten
- verrauschte Daten
- Ausreißer
- inkonsistente Daten

Diese Fehler müssen, falls sie die Analyse mit Data Mining Tools beeinträchtigen könnten vorher entsprechend behandelt werden.

Oft ist es auch nötig einige Daten vorher zu transformieren, da einige Algorithmen für das Data Mining bestimmte Attribute voraussetzen und dadurch insgesamt eine konsistentere Datenbasis geschaffen werden kann.

Die Daten, die für die Analyse mit Data Mining Tools benutzt werden sind nicht immer in einer einzigen Datenquelle verfügbar, sondern sie erstrecken sich über mehrere Datenquellen. Manchmal ist es auch wünschenswert eine neue Variable einzuführen, wie z. B. die Dauer des Auf-

enthalts falls nur das Datum der Aufnahme und Entlassung vorhanden sind. Dafür müssen Methoden im Data Mining Tool vorhanden sein um das zu bewerkstelligen. Das wird häufig durch eine Datenbank oder ein Data Warehouse erledigt in dem die Daten vorhanden sind und durch eine Abfragesprache wie SQL abgefragt werden können.

Oft sind die Daten einfach für eine schnelle Analyse mit Hilfe eines zeitaufwändigem Algorithmus zu umfangreich. Das kann man durch die Bildung von Stichproben, auf denen das Data Mining oder eine vorrangige Analyse angewendet werden soll, erreichen. Dabei kann man ebenso geschichtete Stichproben verwenden um somit die Wahrscheinlichkeit für das Auftreten einer gewissen Untergruppe zu erhöhen. Eine geschichtete Stichprobe wird gebildet, indem man die Daten in m Teilmengen D_i $(1 \leq i \leq m)$ aufteilt um aus jede dieser Teilmengen eine Stichprobe zu ziehen. Die aus der geschichteten Stichprobe resultierende Menge D_S, wird danach wie folgt gebildet, wobei mit S die Ziehung einer Stichprobe dargestellt wird:

$$D_S = \bigcup_{i=1}^{m} S(D_i)$$

Für eine Analyse mit Verfahren des Data Mining sind redundante Attribute in vielen Fällen ein Hindernis. Durch diese können manche Algorithmen schlechtere Regeln erzeugen und eine Analyse dauert teilweise erheblich länger.

Anbei soll hierzu eine Übersicht über die gängigsten Verfahren dargestellt werden. Dabei werden diese Verfahren untersucht, um festzustellen für welche Attributtypen diese nützlich sind und was man dabei beachten muss.

Fehlende Daten

Bei fehlenden Daten muss man beachten, dass manche Daten bewusst fehlen und eine Bedeutung haben können. Um diese korrekt behandeln zu können, sollten diese fehlenden Daten durch einen aussagekräftigen Wert ersetzt werden. Das ist z. B. bei numerischen Werten der Fall, die nur eingetragen wurden, falls tatsächlich ein Wert anfiel. In diesem Falle kann man einen festen Wert oder den Durchschnitt eintragen. Bei kategorischen Variablen wird im Gegensatz dazu oft die am meisten auftretende Klasse, der Modus, eingesetzt. Falls nur wenige Werte fehlen kann man die fehlenden Werte auch manuell eintragen.

Diskretisierung

Bei einer Diskretisierung wird ein numerisches Attribut in ein kategorisches Attribut transformiert. Dafür werden Dabei ist es von entscheidender Bedeutung welcher Wertebereich der Variable in welche Klasse eingeteilt wird. Für die Entscheidung welche Grenzen gewählt werden sollen gibt es viele Möglichkeiten. Man kann z. B. jeder Klasse alle Werte, die in einem Bereich liegen, zuweisen. Falls die Längen der Intervalle für alle Klassen gleich sind, so wird diese Diskretisierung auch als equi-width binning bezeichnet. Die Anzahl an Klassen muss allerdings vorher festgelegt werden.

Die Grenzen können auch so gewählt werden, dass jede Klasse gleich viele Werte enthält. Das wird als equi-depth bezeichnet. Dadurch kann man die Menge des Auftretens für einen Wert besser in die Analyse einbeziehen.

Normalisierung

Oft ist es nötig eine Normalisierung durchzuführen. Das ist vor allem dann nötig, falls man verschiedene numerische Variablen miteinander vergleichen möchte. Das Problem damit ist, dass der Wertebereich der Variablen oft nicht übereinstimmt.

Dazu zählt u. a. auch die min-max-Normalisierung. Damit kann man eine numerische Variable ohne Verlust in den Bereich $[0,1]$ transformieren:

$$X_{neu} = \frac{X_{alt} - \min(X_{alt})}{\max(X_{alt}) - \min(X_{alt})}$$

Das Problem dabei ist, dass Ausreißer einen sehr starken Einfluss ausüben. Bei der sogenannten z-score Normalisierung wird im Gegensatz dazu eine Standardisierung der Form

$$X_{neu} = \frac{X_{alt} - \mu}{\sigma}$$

vorgenommen, sodass die Variable zu anderen numerischen Variablen mit einem anderen Mittelwert / Varianz vergleichbar ist. Dabei wird das Attribut so transformiert, dass die neue Variable den Mittelwert 0 und die Standardabweichung 1 besitzt.

Ausreißer

Ausreißer kann man nach einer vorangegangenen Normalisierung einfach erkennen indem man alle Werte die außerhalb einer gewissen Grenze L liegen, wobei L die Anzahl der Standardabweichungen darstellt, identifiziert. Dabei wird L oft auf 3 gesetzt und die Werte die außerhalb von $[-L, L]$ liegen werden entfernt oder durch einen sinnvollen Wert wie

L oder $-L$ ersetzt. Man kann Ausreißer auch erkennen, indem man sich die Daten visualisiert und dadurch Abweichungen feststellt.

Redundante Attribute

Redundante Attribute verschlechtern die Ergebnisse der Data Mining Analyse teilweise erheblich. Diese Attribute kann man durch Unabhängigkeitstests oder anderen Methoden, wie der Methode des Minimum-Description-Length, auswählen. Bei der Methode des Minimum-Description-Length werden die Attribute basierend auf dem Prinzip von Ockhams Skalpell, wobei eine einfachere Theorie, komplexeren Theorien vorzuziehen ist.

2.5 Problemtypen

Die Methoden des Data Mining kann man grundsätzlich in verschiedene Ebenen aufteilen. In der Literatur gibt es dazu Methoden die mit 1, 2, oder 3 Ebenen arbeiten. Eine Aufzählung dazu finden man in (Alpar & Niedereichholz, 2000) S. 9.

In dieser Arbeit wird im Folgenden eine Herangehensweise mit 3 Ebenen dargestellt. In der ersten Ebene wird das primäre Ziel des Data Mining dargestellt. Dabei kann man zwischen der Prognose und der Beschreibung unterschieden. Bei der Prognose werden Werte von bestimmten Variablen vorhergesagt. Bei der Beschreibung wird hingegen nach Mustern in den vorhandenen Daten gesucht, die vom Menschen verständlich sind. Dabei kann man die Beschreibung auch teilweise für eine Prognose verwenden. Auf der zweiten Ebene folgen die Problemtypen, die im weiteren Verlauf der Arbeit näher beschrieben werden. Schließlich folgen auf der dritten und letzten Ebene die Algorithmen mit denen man die

Problemtypen aus der Ebene 2 lösen kann. Diese werden zu den jeweiligen Problemtypen vorgestellt. vgl. (Fayyad et al., 1996) S. 12

Laut Han muss man zwischen erklärenden und voraussagenden Methoden und Algorithmen unterscheiden. vgl. (Han & Kamber, 2001)

Beim Data Mining kann man eine Reihe von Problemtypen unterscheiden mit denen man unterschiedliche Aufgabenstellungen bearbeiten kann. Da die Algorithmen und Problemtypen im Data Mining im Großen und Ganzen aus Lernmethoden des „Maschinellen Lernen“ bestehen kann man zwischen „überwachten“ und „nicht überwachten“ Lernverfahren unterscheiden.

Bei den „überwachten Lernverfahren“ werden die Daten in zwei Datenmengen aufgeteilt. Eine Trainingsdatenmenge anhand derer das Modell trainiert bzw. erstellt wird und eine Testdatenmenge mit derer Hilfe das Modell auf seine Genauigkeit hin untersucht werden kann.

Voraussagende Verfahren sind hauptsächlich dann von Interesse, wenn der Benutzer eine Prognose bezüglich gewisser Attribute von bestimmten Datensätzen haben möchte. Erklärende Verfahren versuchen aus den Daten, auf denen das Data Mining angewendet wird, vom Benutzer verständliche Regeln zu erhalten. Dabei ist zu beachten dass diese Regeln auch zur Prognose von bisher unbekannten Klassen zu Rate gezogen werden können.

Im Laufe der Arbeit werden die folgenden Problemtypen näher beschrieben:

- Klassifikation
- Prognose / Regression

- Segmentierung / Clustering
- Abweichungsanalyse
- Assoziationsanalyse

Clustering

Clustering wird in der Literatur auch Segmentierung genannt. vgl. (Alpar & Niedereichholz, 2000) S. 9 Mit Hilfe von Clustering kann man eine bestehende noch nicht in Klassen aufgeteilte Menge von Objekten in natürliche Gruppen, auch Cluster genannt, aufteilen. Deshalb sollte es vor allem dann angewendet werden, wenn die Daten noch nicht klassifiziert wurden. Ziel ist es, eine Segmentierung so anzuwenden, dass die Objekte eines Clusters untereinander ähnlich sind und die Objekte verschiedener Cluster untereinander unähnlich sind. (vgl. Han & Kamber, 2001 S. 335) Clustering wird in der Terminologie des Maschinellen Lernens hauptsächlich als nicht überwachtes Lernverfahren angesehen. Es gibt aber auch Algorithmen für das Clustering, die in einem gewissen Maße eine automatische Auswertung zulassen. Diese Verfahren ersetzen jedoch nicht eine Auswertung durch Experten.

Typische Beispiele für das Clustering sind:

- Marktsegmentierung und
- Kundensegmentierung.

Bei der Kundensegmentierung werden die Kunden eines Unternehmens in homogene Gruppen aufgeteilt, um somit speziell zugeschnittene Marketingprogramme auf diese Gruppen anwenden zu können. (vgl. Alpar & Niedereichholz, 2000, S. 121)

Klassifikation

Anders als beim Clustering sind bei der Klassifikation die Anzahl der Klassen, sowie die bereits klassifizierten Fälle vorher bekannt. Ziel bei der Klassifikation ist es dabei, möglichst genaue Klassifikationsregeln für die Fälle zu finden. Somit kann man evtl. nicht bekannte Einsichten in die Daten erhalten. Man kann diese Regeln danach auch für Daten, die noch nicht klassifiziert wurden, anwenden, um damit diese Fälle möglichst gewinnbringend klassifizieren zu können. Die Klassifikation gehört zu den überwachten Lernverfahren.

Typische Beispiele für die Klassifikation sind:

- Klassifikation von „guten“ und „schlechten“ Kunden bezüglich des Umsatzes,
- Klassifikation von Kunden die nach Versendung eines Gutscheins etwas bestellen oder nicht.

Prognose

Bei der Prognose werden numerische Werte anhand von historischen Werten oder anderen Attributen aus der gleichen Zeitperiode vorhergesagt. Dabei sind keine vorher definierten Klassen notwendig. Eine Klassifikation kann aber auch zur Prognose eingesetzt werden. Dabei werden jedoch nur Klassen vorhergesagt, die eine begrenzte Anzahl von Ausprägungen haben können was einem numerischen Wert in der Regel widerspricht.

Assoziationsanalyse

Bei den bisher betrachteten Problemtypen wurden Fälle oder eine Teilmenge der Fälle miteinander verglichen. Die Assoziationsanalyse wird

im Gegensatz dazu benutzt um Abhängigkeiten zwischen Attributen herauszufinden. Das klassische Beispiel für die Assoziationsanalyse ist die Warenkorbanalyse. Dabei werden die Warenkörbe, also die gekauften Produkte der Kunden, in einem Supermarkt untereinander verglichen um Auffälligkeiten beim Kaufverhalten festzustellen.

Abweichungsanalyse

Bei der Abweichungsanalyse werden Fälle untersucht, die vom normalen Verhalten abweichen. Man kann z. B. bei Kreditkartenabrechnung Abweichungen feststellen und somit einen Betrug aufdecken.

2.6 Algorithmen

In diesem Unterkapitel werden einige gängige Data Mining Algorithmen vorgestellt, mit deren Hilfe man die vorgestellten Problemtypen lösen kann. Dabei wird bewusst eine Vorauswahl bei den Algorithmen vorgenommen, da sonst der Rahmen dieser Arbeit gesprengt werden würde. Dabei kann man auf verschiedene Algorithmen aus dem Bereich des Maschinellen Lernens, der Optimierung, der Statistik und einigen anderen Fachgebieten zurückgreifen.

2.6.1 Algorithmen zur Klassifikation

Klassifikation mit Support Vector Machines

Support Vector Machines (SVM) basieren auf der Vapnik-Chervonenkis Theorie. Sie wurden von Vladimir Vapnik entwickelt. Support Vector Machines arbeiten nach dem Prinzip der linearen Diskriminierung von 2 Klassen. Einer positiven und einer negativen Klasse. Dabei wird nach ei-

ner maximal trennenden Hyperebene gesucht. An dem folgenden Beispiel (siehe Abbildung 2) kann man sehen was das bedeutet.

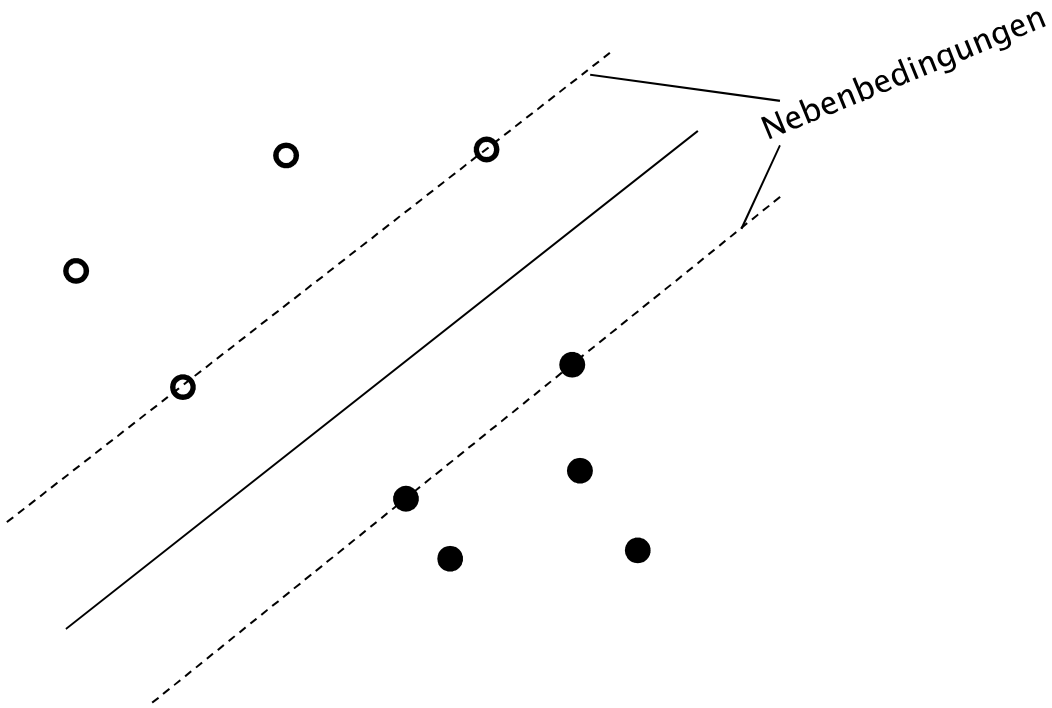

Abbildung 2: Support Vector Machine

Die nicht ausgefüllten Kreise in Abbildung 2 sind die Vektoren die, die positive Klasse +1 repräsentieren. Man kann sehen, dass sie über eine maximal trennende Hyperebene von den Vektoren der negativen Klasse -1 abgegrenzt sind. Die Vektoren, die in Abbildung 2 auf den gestrichelten Linien liegen werden Support Vectors genannt. Diese definieren die maximal trennende Hyperebene eindeutig. Dabei wird die Hyperebene so gewählt, dass sie von den Support Vectors einen maximalen Abstand besitzt. Das wird so gemacht um einen möglichen „Overfit“ an die Trainingsdaten zu verhindern. „Overfit“ bedeutet, dass das Modell zu genau an die Trainingsdaten angepasst wurde. Auf andere Daten, die man mit dem Modell klassifizieren möchte, ist dieses Modell dann nur bedingt anwendbar, da das Klassifikationsmodell zu speziell ist.

Mit $y_i \in \{-1,1\}, 1 \leq i \leq l$ werden die Klassen -1 oder 1 für die l verfügbaren Fälle $x_i \in R^d, 1 \leq i \leq l$ mit d Attributen dargestellt. Die Nebenbedingungen in Abbildung 2 können durch folgende Gleichungen charakterisiert werden:

$$x_i \cdot w + b \geq +1, f\ddot{u}r\ y_i = +1$$

$$x_i \cdot w + b \leq -1, f\ddot{u}r\ y_i = -1$$

Das kann man folgendermaßen umschreiben:

$$y_i(x_i \cdot w + b) \geq +1$$

Man kann zeigen, dass der Abstand zwischen den beiden Hyperebenen die die Nebenbedingungen darstellen $\frac{2}{\|w\|}$ beträgt. Da es im Allgemeinen mehrere trennende Hyperebenen gibt muss die Hyperebene gefunden werden, die die Daten am besten trennt.

Die Optimierungsaufgabe lautet wie folgt:

Minimiere $\|w\|^2$ bzgl. den Nebenbedingungen

$$y_i(x_i \cdot w + b) \geq +1, 1 \leq i \leq l$$

Für diese konvexe Optimierungsaufgabe sind bereits Algorithmen bekannt, die die optimale Lösung finden können. vgl. (Burges, 1997) S. 8-9, vgl. (Alpaydin, 2004) S. 218-221

Diese Methode funktioniert jedoch nur bei linear trennbaren Daten. Falls man keine linear trennbaren Daten vorliegen hat kann man durch die Einführung von Schlupfvariablen gewisse Fehler zulassen. Man kann auch nicht lineare Trennfunktionen erzeugen. Diese funktionieren über eine Transformation des Vektorraums in einen höherdimensionalen Vektorraum. In diesem Vektorraum sind die Fälle dann wieder linear

trennbar. Nach einer Rücktransformation in den ursprünglichen Vektorraum ist die Entscheidungsgrenze danach nicht linear.

Ein Nachteil dieser Methode ist, dass in dieser einfachen Form nur zwei Klassen klassifiziert werden können. Deshalb wurden im Laufe der Zeit Erweiterungen vorgenommen mir deren Hilfe man eine Klassifikation mit mehreren Klassen durchführen kann. Eine Möglichkeit das umzusetzen funktioniert über mehrere Support Vector Machines, die dann mittels Mehrfachentscheidung die endgültige Klassifikation ermitteln.

Klassifikation mit Entscheidungsbäumen

Entscheidungsbäume können zur Klassifikation und zur Prognose benutzt werden. Wobei hier nur auf die Klassifikation mit Entscheidungsbäumen näher eingegangen wird.

Mit Hilfe von Entscheidungsbäumen kann man anschauliche, für den Menschen verständliche Regeln für eine Klassifikation erzeugen. Entscheidungsbäume bestehen aus Knoten, die hierarchisch aufeinander aufgebaut sind. Der erste Knoten des Entscheidungsbaums wird Wurzel genannt. In der Wurzel sind alle Fälle, die zur Klassifikation benötigt werden (Trainingsmenge), vereint. Beginnend bei der Wurzel werden die Fälle, die für einen Knoten zur Verfügung stehen bezüglich einer Entscheidungsregel in zwei oder mehr disjunkte Teilbereiche, oder auch Unterbäume aufgeteilt. Dieser Vorgang wird auch Splitting genannt. Die in der Praxis genutzten Algorithmen führen meist binäre Splits durch. Dafür werden die Attribute, die in einem Knoten zur Verfügung stehen, bezüglich einer Splittingstrategie ausgewählt. Ein Knoten wird Blatt genannt, falls ein weiteres Splitting nicht mehr weiter durchgeführt werden soll.

Die gängigsten Algorithmen für die Entscheidungsbaumerstellung sind u. a. CART (Classification and Regression Trees), C4.5, ID3 (Iterative Dichotomiser 3) und CHAID (Chi-square Automatic Interaction Detectors).

Im Laufe der letzten Jahrzehnte haben sich eine Reihe Entscheidungsbaumalgorithmen etabliert, wobei hier nur die grundlegenden Probleme und Ideen bei Entscheidungsbaumalgorithmen aufgezeigt werden.

Hier soll nun der ID3-Algorithmus vorgestellt werden.

Algorithmus (ID3) vgl. (Han & Kamber, 2001) S. 285

ID3(Fälle, Liste der verfügbaren Attribute L)

1. erstelle eine Wurzel R
2. falls alle Fälle mit genau einer Klasse C klassifiziert wurden, gibt R mit der Bezeichnung der Klasse C zurück
3. falls L leer ist, gibt R mit der Bezeichnung der Klasse C_{max} zurück, die am meisten in der Menge der Fälle auftritt
4. wähle $A \in L$ als das Attribut aus, das die Menge der Fälle „am besten“, bzgl. der Splittingstrategie klassifiziert
5. markiere den Knoten R mit dem Attribut A
6. $\forall\, a_i \in A$ (Partitionierung der Menge)
    a. füge einen Ast an R an und benenne diesen Ast mit „$A = a_i$“
    b. sei s_i die Menge aller Fälle bei denen $A = a_i$ gilt
    c. falls s_i leer ist, setze ein Blatt an den Ast und markiere dieses Blatt mit der am meisten auftretenden Klasse C_{max} aus der Menge der Fälle
    d. sonst füge den Knoten den ID3($s_i, L \setminus A$) zurückgibt an den Ast an

„Die Splittingstrategie ist der Kern eines Entscheidungsbaum-Klassifikators. „ (Ester & Sander, 2000) S. 127 Dabei muss man zwischen einem Split für kategorische und numerische Variablen unterscheiden. Es lässt sich bei numerischen Attributen mithilfe der möglichen Vergleichsoperationen $\leq$, $<$, $\geq$ und $>$ ein Splitting durchführen. Bei kategorischen Attributen kann man nur darstellen ob ein Wert des Attributes innerhalb einer Menge mit allen Werten dieses Attributs liegt.

Für Entscheidungsbäume ist es entscheidend, eine geeignete Metrik für das Splitting zu finden. Dabei wird versucht herauszufinden, welches Attribut den Attributraum bezüglich der Reinheit am besten aufteilt. Eine große Reinheit bedeutet für einen Knoten in einem Entscheidungsbaum, dass die Klassifikationsgenauigkeit bezüglich einer Klassifikation des Knotens möglichst hoch ist. Verschiedene Algorithmen definieren dafür unterschiedliche Möglichkeiten. Hier werden einige Beispiele aufgezeigt.

Wir definieren mit T die Menge aller Trainingsobjekte. Bei einem m-ären Split bezüglich eines Attributs wird diese Menge in m disjunkte Teilmengen T_1, T_2, …, T_m aufgeteilt. Mit p_i wird die relative Häufigkeit der Klasse c_i in T bezeichnet.

Eine sehr einfache Splittingregel benutzt den mittleren Informationsgehalt für verschiedene Attribute. Dieser Wert wird auch Entropie genannt. Die Entropie wird folgendermaßen definiert:

$$Entropie(T) = -\sum_{i=1}^{n} p_i \cdot log_2(p_i)$$

Mit Hilfe der Entropie kann man den Informationsgewinn für ein Attribut A bestimmen:

$$Informationsgewinn(T, A) = Entropie(T) - \sum_{i=1}^{m} \frac{|T_i|}{|T|} \cdot Entropie(T_i)$$

Das Attribut mit dem höchsten Informationsgewinn wird dann als das nächste Attribut für das Splitting genommen. Der Algorithmus ID3 benutzt den Informationsgewinn als Splitting-Strategie.

Eine weitere Splittingregel ist der sogenannte GINI-Koeffizient. Dieser wird folgendermaßen definiert:

$$GINI(T) = 1 - \sum_{i=1}^{k} {p_i}^2$$

Der GINI-Index gibt an wie groß die Reinheit / Unreinheit der Attribute bzgl. der Klassifikationsvariablen ist. vgl. (Ester & Sander, 2000) S. 128-129

Bei einigen Algorithmen zur Erzeugung eines Entscheidungsbaums kann man ein sogenanntes Pruning durchführen. Das ist eine vielfach angewendete Methode um den Baum zu beschneiden und somit eine höhere Generalisierungsfähigkeit zu erhalten. Dadurch wird zwar der Fehler auf den Trainingsdaten steigen, kann aber den Fehler für eine unabhängige Testmenge erhöhen.

Die Entscheidungsbaumverfahren funktionieren, falls Attribute enthalten sind, die nur scheinbar mit den Daten korrelieren nicht gut. Sie funktionieren mit wenigen aussagekräftigen Attributen am besten. Bei vielen unwichtigen Attributen wird die Suche nach den, am optimalen Entscheidungsbaum beteiligten, Attributen schwieriger.

Naive Bayes

Es gibt eine Reihe von Bayes-Klassifikatoren. Diese basieren auf dem Satz von Bayes aus der Statistik und dementsprechend auf bedingte Wahrscheinlichkeiten. Dabei wird die Klassenzugehörigkeit einer Datenmenge X anhand einer Hypothese vorhergesagt. Hier soll nun der Naive Bayes Klassifikator in Anlehnung an (Han & Kamber, 2001) S. 297-298 dargestellt werden.

Definition (Satz von Bayes)

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

X	*Datenmenge mit unbekannter Klassifizierung,* $X = (x_1, \dots, x_n)$
H	*eine Hypothese, sodass* X *einer Klasse* C *angehört*
$P(H\|X)$	*Wahrscheinlichkeit, dass die Hypothese* H *richtig ist, wenn die Datenmenge* X *gegeben ist*
$P(H)$	*A-priori Wahrscheinlichkeit für die Hypothese*
$P(X)$	*A-priori Wahrscheinlichkeit für die Datenmenge*

Beim Naive Bayes Klassifikator sind wir an der a-posteriori Wahrscheinlichkeit $\mathrm{P(H|X)}$ interessiert. Da H die verschiedenen Klassen repräsentiert, muss man, um einen sinnvollen Klassifikator aus der Datenmenge X zu erhalten, die Klasse finden die $\mathrm{P(H|X)}$ maximiert. Es sei m die Anzahl der Klassen. Dann kann man für jede Klasse $C_i, 1 \leq i \leq m$ die a-posteriori Wahrscheinlichkeit berechnen. Das kann man mit Hilfe des Satzes von Bayes so ausdrücken:

$$\max_i \frac{\mathrm{P(X|C_i)P(C_i)}}{\mathrm{P(X)}}$$

Da dieser Ausdruck über alle Klassen C_i maximiert werden soll, kann man den Nenner wegen einem konstanten Wert für alle Klassen ignorieren. Damit reduziert sich die Formel zu

$$\max_i \mathrm{P}(\mathrm{X}|\mathrm{C_i})\mathrm{P}(\mathrm{C_i})$$

Die $\mathrm{P}(\mathrm{C_i})$ kann man relativ einfach aus den Trainingsdaten schätzen, indem man dafür einfach die relative Häufigkeit der C_i schätzt.

$\mathrm{P}(\mathrm{X}|\mathrm{C_i})$ kann man dagegen nur äußert schwierig berechnen. Beim Naive Bayes wird davon ausgegangen, dass die Attribute untereinander unabhängig sind. Das kann man benutzen um $\mathrm{P}(\mathrm{X}|\mathrm{C_i})$ folgendermaßen umzuschreiben.

$$\mathrm{P}(\mathrm{X}|\mathrm{C_i}) = \prod_{k=1}^{n} \mathrm{P}(x_k|\mathrm{C_i})$$

Diese Wahrscheinlichkeiten können dagegen aus der Datenmenge berechnet werden.

Um schließlich einen unbekannten Fall X zu klassifizieren wird $\mathrm{P}(\mathrm{X}|\mathrm{C_i})P(C_i)$ für jede Klasse C_i berechnet und die Klasse mit dem höchsten Wert wird dem Fall X zugeteilt.

Bei Naive Bayes ist es unbedingt notwendig, dass die Unabhängigkeit für die Daten vorher sichergestellt wird. Ein weiteres Problem ist, dass manche Attribute nicht von der Klassifikationsvariablen abhängen sondern von anderen Variablen und Faktoren die nicht mit erfasst wurden. Der Vorteil für die Naive Bayes Methode ist vorrangig, dass dieser Klassifikator relativ einfach gelernt werden kann. Dadurch ist dieser Algorithmus auch auf sehr große Daten anwendbar. Man kann dadurch die fertige

Klassifikation auch einfacher anhand der bedingten Wahrscheinlichkeiten für die Klassenvariable nachvollziehen.

2.6.2 Algorithmen zur Segmentierung / Clustering

Bei der Segmentierung werden typischerweise die Algorithmen aus der Clusteranalyse angewendet. In diesem Abschnitt sollen einige Clusteranalyseverfahren und die dazugehörigen Klassen von Algorithmen vorgestellt werden.

Um eine Clusteranalyse durchführen zu können ist es sehr wichtig den richtigen Algorithmus für das spezielle Problem anzuwenden. Dabei sollte man zuerst eine geeignete Ähnlichkeitsfunktion wählen. Die Wahl einer geeigneten Ähnlichkeitsfunktion ist vom Attributtyp abhängig. Man kann für numerische Attribute z. B. eine Metrik wählen. Eine Metrik wird folgendermaßen definiert:

Definition: (Metrik)
Eine Funktion $\mathrm{d(x,y)}$ *ist eine Metrik, falls Folgendes gilt:*

1. $d(x,y) = 0$ *genau dann, wenn* $x = y$
2. $d(x,y) \geq 0 \; \forall x,y$
3. $d(x,y) = d(y,x) \; \forall x,y$
4. $d(x,z) \leq d(x,y) + d(y,z) \; \forall x,y,z$

Als Beispiel kann man u.a. folgende Metriken angeben:

Euklidischer Abstand:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Maximums-Metrik:

$$d(x,y) = \max\left(\sum_{i=1}^{n} |x_i - y_i|\right)$$

Allgemeine L_p-Metrik:

$$d(x,y) = \sqrt[p]{\sum_{i=1}^{n} (x_i - y_i)^p}$$

Für kategorische Variablen müssen aber andere Abstandsfunktionen gefunden werden.

Es gibt verschiedene mögliche Klassen von Algorithmen für das Clustering. Man kann u. a. folgende aufzählen:

- partitionierende Verfahren
- hierarchische Verfahren
- probabilistische Verfahren
- dichtebasierte Verfahren
- gitterbasierte Verfahren
- modellbasierte Verfahren

Dabei ist zu beachten, dass es durchaus auch Algorithmen geben kann, die zu mehreren Klassen dazugehören.

In der weiteren Betrachtung sollen einige grundlegende Eigenschaften von einigen Klassen von Algorithmen betrachtet und ein gängiges Verfahren dargelegt werden.

Partitionierende Verfahren

Partitionierende Verfahren teilen die Menge der Objekte in k Cluster auf. Die Anzahl der Cluster k muss dabei vorher festgelegt werden. Dabei

enthält jeder Cluster mind. ein Objekt und jedes Objekt gehört zu genau einem Cluster. vgl. (Ester & Sander, 2000) S. 51

Dafür müssen vorerst Centroide definierte werden.

Definition (Centroide) (Ester & Sander, 2000) S. 55
Ein Centroid μ_C stellt den Mittelwert eines Clusters C folgendermaßen dar:

$$\mu_C = \left(\overline{x_1}(C), \overline{x_2}(C), \ldots, \overline{x_d}(C)\right)$$

Dabei ist $\overline{x_j}(C) = \frac{1}{n_C}\sum_{p\epsilon C} x^p{}_j$ der Mittelwert der j-ten Dimension aller Punkte in C und n_C die Anzahl der Objekte in C.

Definition: (Maß für die Kompaktheit eines Clusters) (Ester & Sander, 2000) S. 55
Summe der quadrierten euklidischen Distanzen zum Centroid:

$$TD^2 = \sum_{i=1}^{k} dist(p, \mu_C)^2$$

Der k-means ist ein sehr weit verbreitetes Verfahren zur Clusteranalyse. Dabei wird die Anzahl der Cluster k vorgegeben und eine passende Abstandsfunktion.

Algorithmus (k-means) vgl. (Han & Kamber, 2001) S. 349

1. k Clusterzentren werden zufällig ausgewählt
2. basierend auf einer Abstandsfunktion werden die Objekte dem Cluster C zugeteilt zu dem der Centroid μ_C den kleinsten Abstand aufweist
3. berechne die Centroide μ_C für jeden Cluster neu

4. falls eine Veränderung bei der Clusterzuordnung der Objekte stattgefunden hat gehe zu Schritt 2

Dabei muss man beachten, dass das Clustering beim k-means stark abhängig ist von der anfänglichen Auswahl der Clusterzentren. Das ist vor allem ein Problem, falls die Cluster zu stark überlappen oder falls sie eine unübliche Form, wie z. B. ein langgezogenes Band besitzen.

Hierarchische Verfahren

Hierarchische Verfahren gliedern sich in agglomerative und divisive Verfahren.

Agglomerative Verfahren beginnen mit mehreren Clustern mit einem Objekt pro Cluster. Basierend auf eine Distanzfunktion zwischen den Clustern werden dabei die ähnlichsten Cluster sukzessive miteinander verbunden. Die bekanntesten hierarchischen Verfahren sind die Single-Link, Average-Link und Complete-Link Algorithmen. Jeder dieser Verfahren besitzt eine andere Abstandsfunktion. Der kleinste Abstand wird beispielsweise beim Single-Link-Verfahren benutzt.

Divisive Verfahren beginnen dagegen mit einem Cluster in dem alle Objekte vorhanden sind und teilen diesen danach in kleinere Cluster auf.

2.6.3 Algorithmen zur Assoziationsanalyse

Es sei $J = \{i_1, \dots, i_m\}$ eine Menge von Items (Artikel) gegeben. Sei D, eine Menge von Transaktionen in der Datenbank, wobei jede Transaktion T eine Menge von Items beinhaltet, sodass $T \subseteq J$ gilt. Jede Transaktion wird durch eine spezielle Variable TID eindeutig identifiziert. Sei A eine Menge von Items. Eine Transaktion T enthält A genau dann, wenn $A \subseteq T$ gilt. Eine Assoziationsregel ist eine Implikation der Form $A \Rightarrow B$, wobei B

analog zu A eine Menge von Items, die disjunkt zu A, ist. vgl. (Han & Kamber, 2001) S. 227-228

Um bei der Assoziationsanalyse sinnvolle Lösungen zu finden müssen Kennzahlen angegeben werden die die Regeln auf ihre Interessantheit und Nützlichkeit untersuchen können. Dafür werden im Folgenden der Support $Supp$, die Konfidenz $Conf$ und der Lift definiert.

Definition (Support) (Han & Kamber, 2001) S. 228
Der Support $Supp$ der Regel $A \Rightarrow B$ ist definiert als der Anteil der Transaktionen in denen A und B vorkommt. Das kann man auch wie folgt, als Wahrscheinlichkeit ausdrücken:

$$Supp(A \Rightarrow B) = P(A \cup B)$$

Definition (Konfidenz) (Han & Kamber, 2001) S. 228
Für die Regel $A \Rightarrow B$ zeigt die Konfidenz $Conf$ an wie oft A zusammen mit B in der Menge aller Transaktionen vorkommt. Das kann man wie folgt als bedingte Wahrscheinlichkeit ausdrücken:

$$Conf(A \Rightarrow B) = P(B|A)$$

Definition (Lift)
Der Lift für die Regel $A \Rightarrow B$ ist folgendermaßen definiert:

$$Lift(A \Rightarrow B) = \frac{Conf(A \Longrightarrow B)}{P(B)}$$

Regeln die einen hohen Support und Konfidenzwert besitzen werden starke Regeln genannt. Mit dem Lift kann etwas über die Bedeutung der Regel aussagen. Dadurch wird der Einfluss von A auf B gemessen.

Einer der bekanntesten Algorithmen für die Assoziationsanalyse ist der Apriori-Algorithmus. Für den Apriori Algorithmus müssen am Anfang

ein minimaler Support $Supp_{min}$ und eine minimale Konfidenz $Conf_{min}$, die die Regeln am Ende haben sollten, festgelegt werden.

Ein Itemset I ist eine Menge von Items. Ein k-Itemset besitzt k Items. Ein häufig auftretendes Itemset besitzt einen Support der größer ist als $Supp_{min}$.

Algorithmus (Apriori)

1. finden aller häufig auftretenden Itemsets
2. Erzeugen von starken Assoziationsregeln aus den häufig auftretenden Itemsets

Die häufig auftretenden Itemsets kann man sukzessive finden indem man startend bei $k = 1$ die häufig auftretenden k-Itemsets durch Hinzufügen eines weiteren Items in $(k + 1)$-Itemsets transformiert. Aus diesen werden danach nur die häufig auftretenden Itemsets herausgefiltert, um diese danach weiter zu untersuchen bis keine häufig auftretenden k-Itemsets mehr zur Analyse vorhanden sind.

Zur Erzeugung der starken Assoziationsregeln werden aus allen häufig auftretenden Itemsets I alle mögl. Teilmengen gebildet. Aus jeder Teilmenge $S \in I$ wird die Regel $S \Rightarrow (I \setminus S)$ gebildet. Es werden nur Regeln weiter betrachtet bei denen $\frac{Supp(I)}{Supp(S)} \geq Conf_{min}$ gilt. vgl. (Han & Kamber, 2001) S. 227-228

2.7 Verifizierungsmethoden

Beim Data Mining gibt es eine Reihe von Verifizierungsmethoden mit denen man überprüfen kann, ob die gefundenen Muster und Informationen plausibel sind.

„Erst durch die Auswertung erhält das Data Mining wirklich einen Sinn. „ (Witten & Frank, 2001) S. 127

Beim „überwachten Lernen“, das bereits dargestellt wurde, gibt es eine Trainingsdatenmenge und eine Testdatenmenge. Für die Auswertung sollte man möglichst die Testdatenmenge nehmen, da ein Überprüfen mittels der Trainingsdaten keine Aussage über die Klassifikationsgenauigkeit bezüglich der Anwendung des Algorithmus auf neue Daten zulässt. Nichtdestotrotz ist es manchmal nützlich den Klassifikationsfehler bezüglich der Trainingsdatenmenge zu kennen. Dieser wird auch als Resubstitutionsfehler bezeichnet. Dadurch kann man u.a. erkennen, inwiefern der Klassifikationsalgorithmus überhaupt die Trainingsdatenmenge korrekt erfasst hat. Bei sehr großen Datenmengen kann man einfach aufgrund der großen Anzahl an Fällen eine Trainingsdatenmenge und eine Testdatenmenge zufällig erstellen und das Modell, das anhand der Trainingsdatenmenge erstellt wurde, mithilfe der Testdatenmenge auf Fehler überprüfen. Dieses Verfahren wird auch Holdout-Methode genannt. vgl. (Witten & Frank, 2001) S. 127

Oft ist es aber der Fall, dass nur sehr wenige Daten wirklich für das Data Mining in Frage kommen. Diese Situation kann z. B. bei Klassifikationsproblemen auftreten, da die Daten unter Umständen aufwändig zu klassifizieren sind. Deshalb muss man Techniken finden, um trotzdem sinnvolle Muster, die statistisch signifikant sind, aus den Daten herauszufil-

tern. Eine dieser Techniken ist die k-Kreuzvalidierung. Darüber hinaus sind statistische Test sinnvoll, die verschiedene Modelle auf ihre Signifikanz überprüfen können. vgl. vgl. (Witten & Frank, 2001) S. 127

Die k-fache Kreuzvalidierung ist sinnvoll falls die Menge der Daten begrenzt ist. Dabei wird die Datenmenge vorher in k gleichgroße Partitionen aufgeteilt. Danach wird das Data Mining k-mal mit einer Auswahl aus (k-1) Partitionen für das Training und einer Partition für das Testen durchgeführt. Für die k Fehlerraten der Testdatenmenge wird anschließt der Durchschnitt gebildet um eine Abschätzung für den Gesamtfehler zu erhalten. vgl. (Mitchell, 1997) S. 112

Kombination mehrerer Modelle

Oftmals ist es nötig, mehrere Modelle zur Entscheidungsfindung zu kombinieren um ein besseres Modell zu finden. Dafür werden in dieser Arbeit zwei Möglichkeiten dargestellt. Bagging, kurz für „bootstrap aggregation“, und Boosting. Sie basieren im Grunde auf der gleichen Idee, nämlich der Mehrheitsentscheidung. Bei einer Mehrheitsentscheidung wird nicht nur ein Klassifikator herangezogen, sondern es werden mehrere Klassifikatoren betrachtet. Man kann diese Methoden aber auch für Prognoseprobleme verwenden, falls eine numerische Variable geschätzt werden soll. Es ist aber nötig, dass der Fehler des Modells abgeschätzt werden kann.

Bagging läuft folgendermaßen ab. Sei S eine Menge von s Fällen und T die maximale Anzahl an Iterationen. Für die Iteration $t \leq T$ wird eine Trainingsmenge S_t aus der Menge S durch Ziehen mit Zurücklegen gebildet. Für die Trainingsmenge S_t wird ein Klassifikator C_t gelernt. Falls man nun einen noch nicht klassifizierten Fall klassifizieren möchte so wird für jeden Klassifikator C_t, $1 \leq t \leq T$, eine Klasse vorhergesagt. Der

Fall wird schließlich mit der Klasse klassifiziert die am meisten von den T Klassifikatoren gewählt wird. vgl. (Han & Kamber, 2001) S. 324-325

Beim Boosting werden die Fälle aus einer Trainingsmenge T im Gegensatz dazu mit $w_i, 1 \leq i \leq |T|$, gewichtet. Nachdem der Klassifikator C_m in der Iteration $m,\ 1 \leq m \leq M$, gelernt wurde, werden die w_i so verändert, dass nicht korrekt klassifizierte Fälle mit einem höheren Gewicht versehen werden. Die Damit kann man diese durch einen weiteren Klassifikator in eine der nächsten Iterationen besser klassifizieren. Ebenso wird ein Gewicht α_m für den Klassifikator C_m, so gebildet, dass Klassifikatoren mit einer höheren Klassifikationsgenauigkeit höher gewichtet werden. Dadurch werden die besseren Klassifikatoren bevorzugt. Bei einem noch nicht klassifizierten Fall wird demnach die Klasse zugewiesen die mit Hilfe einer gewichteten Mehrheitsentscheidung der einzelnen Klassifikatoren C_m am häufigsten auftritt. Dabei werden die Gewichte α_m der Klassifikatoren C_m zur Entscheidungsfindung benutzt. vgl. (Han & Kamber, 2001) S. 324-325, (Hastie, Tibshirani, & Friedman, 2001) S. 299-302

3 Oracle Data Mining - Funktionalitäten

In diesem Kapitel werden die Funktionalitäten des Oracle Data Mining dargestellt. Es werden die Möglichkeiten der Durchführung der Analyse aufgezeigt. Des Weiteren werden einige Tools und Interfaces, die das Data Mining unterstützen, vorgestellt. Die implementierten Algorithmen werden anhand der Funktionalitäten im Oracle Data Miner untersucht und dargestellt.

3.1 Grundlagen von Oracle Data Mining

Oracle Data Mining ist eine Zusatzoption der Oracle Database 11*g* Enterprise Edition. (vgl. Oracle Corporation, 2010) Mit Oracle Data Mining kann man Data Mining Aufgaben mit Hilfe der Oracle Datenbank durchführen.

Man hat verschiedene Möglichkeiten Oracle Data Mining in Produktivsysteme einzubinden. Dabei stellt der Oracle Data Miner, eine graphische Benutzeroberfläche für Oracle Data Mining, den Standardzugriff dar. Im weiteren Verlauf der Arbeit soll die Nutzung des Oracle Data Miners beschrieben werden um dann im nächsten Kapitel Aufgaben damit lösen zu können. Über die grafische Benutzeroberfläche hinaus werden einige Programmierinterfaces für das Data Mining bereitgestellt. Das ist vor allem dann sinnvoll wenn man ein Tool für das Data Mining entwickeln will, um nicht auf den Oracle Data Miner angewiesen zu sein. Des Weiteren sind diese Programmierinterfaces nützlich, falls die Funktionalitäten des Oracle Data Miners nicht ausreichen und man neue Modelle, Module oder Auswertungsmöglichkeiten hinzufügen möchte.

Insgesamt gibt es drei Programmierinterfaces die von Oracle für das Oracle Data Mining zur Verfügung gestellt werden:

- Java Interface,
- PL/SQL Interface,
- R Interface.

Die Java und PL/SQL Interfaces kann man verwenden, falls man Data Mining in ein Projekt einbinden möchte, in dem einer dieser beiden Sprachen benutzt wird. Das R Interface ist nützlich für Anwender, die mit der Programmiersprache R für statistische Anwendungen besser vertraut sind. Im weiteren Verlauf der Arbeit konzentrieren wir uns nur auf den Oracle Data Miner.

Man kann auch „Predictive Analytics“ für ein automatisiertes Data Mining verwenden. Dafür existieren die folgenden drei Methoden:

- Predict
- Explain
- Profile

Für das „Predictive Analytics“ ist auch ein Excel-Addin verfügbar.

3.2 Beschreibung des Oracle Data Miners

Der Oracle Data Miner ist eine Erweiterung des SQL Developer 3. In diesem Unterkapitel werden die implementierten Algorithmen des Oracle Data Miners aufgezeigt. Zusätzlich werden die Möglichkeiten und Arbeitsschritte kurz umrissen.

Systemvoraussetzungen

Für das Oracle Data Mining und den Oracle Data Miner benötigt man die Oracle Database 11g Enterprise Edition mit Oracle Data Mining Option. Um die Data Mining Funktionalitäten benutzen zu können, muss zunächst das Data Mining Repository installiert werden. Dafür stehen zwei Methoden zur Verfügung, die beide für die Installation des Data Mining Repository benötigt man DBA-Rechte für die Datenbank voraussetzen. Bei der ersten Methode werden einige Scripts benötigt, die man auf der Website von Oracle herunterladen kann. Diese werden dann nach dem Ablaufplan in der Readme-Datei „install_scripts_readme.html“ installiert. Dabei bestand allerdings das Problem, dass man danach das Repository nicht für das Data Mining benutzen konnte. Deshalb wurde die zweite Methode benötigt, um das Data Mining Repository zu installieren. Dafür muss man zuerst ein evtl. schon installiertes Repository entfernen. Das kann über ein Script, das bei den Scripts aus der ersten Methode vorhanden ist, erledigt wird. Die zweite Methode funktioniert über das Interface des Oracle Data Miner. Dabei wird über den Navigator des Oracle Data Miner eine Verbindung ausgewählt mit der man das Oracle Data Mining durchführen will. Beim Erstellen eines Projektes erscheint eine Abfrage ob man das Data Mining Repository installieren möchte, falls es noch nicht installiert wurde.

Grafische Benutzeroberfläche

Hier wird die grafische Benutzeroberfläche des Oracle Data Miners beschrieben. Für die Durchführung des Data Mining wird der Data Miner Navigator verwendet. Diesen kann man über Tools → Data Mining → Show Navigator einblenden. In diesem Navigator muss man zuerst eine Verbindung zum Schema der Datenbank anlegen mit dem man Data Mi-

ning durchführen will. In diesem Schema ist es danach möglich, ein neues Data Mining Projekt anzulegen. In diesem Projekt kann man schließlich Workflows erstellen. Der Data Miner Navigator wird in Abbildung 3 dargestellt.

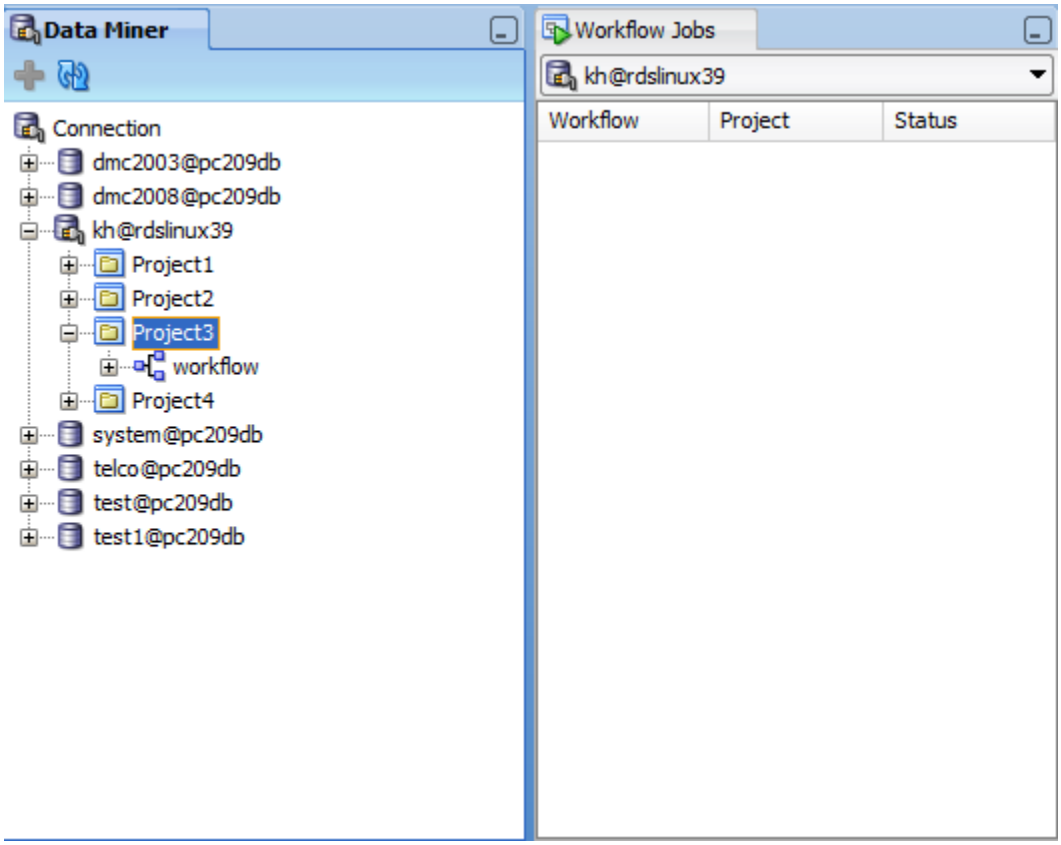

Abbildung 3: Data Miner Navigator

In einem Workflow kann man Knoten für das Data Mining anlegen. In Abbildung 4 kann man einen Beispielworkflow sehen.

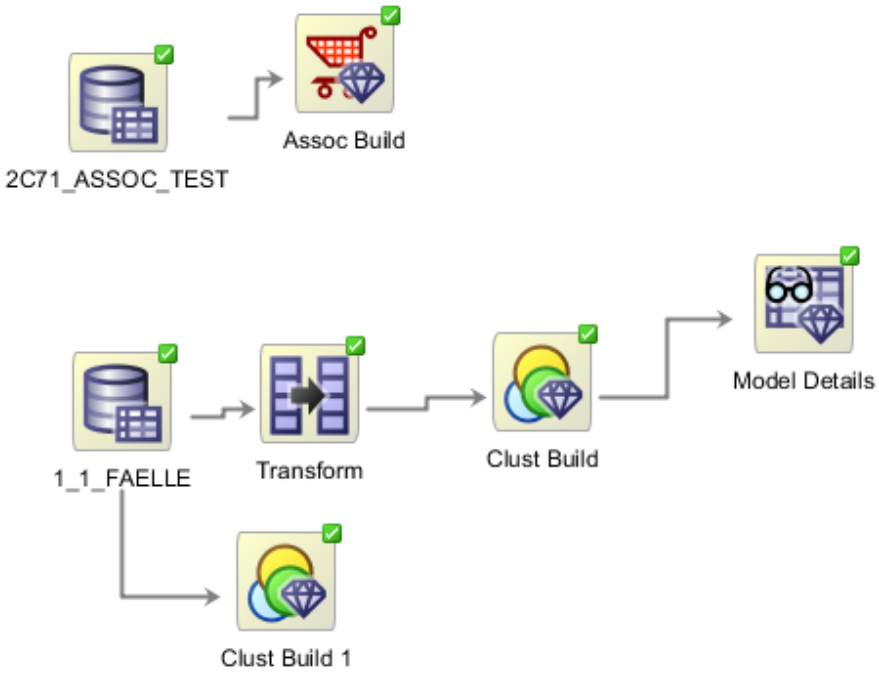

Abbildung 4: Workflow-Aufbau

Es soll eine Übersicht über die Knoten dargestellt werden. Die Knoten kann man über die „Component Palette“ auswählen und auf den Workflow ablegen. Dabei kann man aus verschiedenen Bereichen wählen. Unter Models kann man u. a. die Problemtypen des Data Mining auswählen die man zur Analyse benutzen möchte. Dabei kann man auch, mit Hilfe von „Model Details“ Daten aus den erstellten Modellen erhalten. Das könnten z. B. Regeln, Bäume, Cluster, … sein. Unter „Evaluate and Apply“ finden sich Knoten zum Testen der Modelle anhand von bereitgestellten Testdaten und Knoten zum Anwenden der Regeln auf neue Daten wieder. Der nächste Bereich „Data“ beinhaltet vor allem Knoten mit denen man Tabellen/Views erstellen/einlesen/updaten kann. Mit Hilfe von „Explore Data“ kann man sich die Attribute von Tabellen/Views als Histogramme anzeigen lassen. Unter „Transforms“ kann man die Daten geeignet transformieren / aufbereiten. Diese Methoden werden im nächsten Unterkapitel dargestellt. Die „Component Palette“ mit den „Models“ kann man in Abbildung 5 sehen.

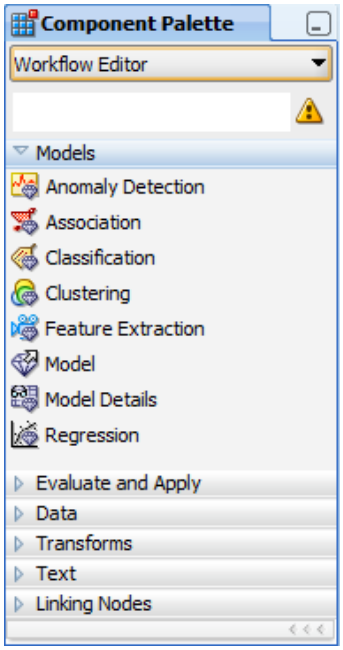

Abbildung 5: Component Palette mit Models

3.3 Methoden zur Datenvorbereitung / Datenaufbereitung

Mit Oracle Data Mining können Daten manuell und automatisch vorbereitet werden. Dabei muss man bei der automatischen Datenaufbereitung zwischen Methoden, die im Algorithmus implementiert sind, die abschaltbar und nicht abschaltbar sind, unterscheiden.

Im Folgenden wird ein Überblick über die manuellen Möglichkeiten der Datenvorbereitung dargestellt. Die automatischen Datenvorbereitungen sind von Algorithmus zu Algorithmus unterschiedlich. Einen Überblick dazu gibt es in den Concepts zu Oracle Data Mining. (siehe (Oracle Corporation, 2010))

Folgende Möglichkeiten stehen zur Vorbereitung der Daten zur Verfügung:

- Aggregation,
- Filter Columns,
- Filter Columns Details,
- Filter Rows,
- Join,
- Sample,
- Transform.

Mit Hilfe von Aggregate kann man transaktionale Attribute in ein neues Attribut zusammenfassen / aggregieren. Dabei kann man, abhängig vom Datentyp, im Oracle Data Miner verschiedene Methoden benutzen. Für numerische Variablen kann man u. a. die Summe, den Durchschnitt, das Minimum, Maximum oder die Anzahl berechnen. Für kategorische Variablen kommen im Gegensatz dazu nur wenige Methoden zur Anwendung. Für diese Variablen kann man u. a. den Modalwert, oder die Anzahl berechnen. Um diese Methoden anwenden zu können, muss allerdings vorher eine group-by-Klausel definiert werden, wonach die Attribute aggregiert werden. Dazu kann in den einzelnen Aggregationen eine sub-group-by Klausel bestimmet werden. Ohne eine sub-group-by Klausel ist die Ausgabe ein normales numerisches / kategorisches Attribut. Falls aber das Gegenteil der Fall ist, wird das Ausgabeattribut einen sogenannten nested, also verschachtelten, Datentyp besitzen. Dabei existieren „Nested Numerical“ und „Nested Categorical“ als die zwei verschiedenen nested Datentypen.

Mit Hilfe von Filter Columns kann man Attribute bezüglich geeigneter Kriterien herausfiltern. Dabei kann man grundlegend die Qualität der Daten bzgl. des Anteils an fehlenden, konstanten, und verschiedenen Werten messen. Des Weiteren kann man die Wichtigkeit der Attribute

bzgl. einer Zielvariablen bestimmen. Durch diese Methoden kann man Attribute, die wenig Aussagekraft für die Aufgabenstellung besitzen, vorher entfernen. Die Attribute müssen aber manuell herausgefiltert werden, da diese Filter nur als Hilfsmittel zur Wahl der Attribute dienen. Diese Gewichtungen der Attribute können durch „Filter Columns Details“ dargestellt werden.

Mit Hilfe von „Filter Rows“ kann man einige Fälle / Zeilen mit Hilfe eines Filters entfernen. Dies ist analog zu der where-Klausel bei einer select Anweisung.

Den „Join“-Knoten kann man benutzen, um zwei oder mehrere Knoten miteinander zu verbinden. Dabei kann man analog zu den klassischen SQL-Joins „Inner“, „Outer“ und „kartesische Joins“ erstellen.

Mit „Sample“ kann man eine Stichprobe aus den Daten erstellen, falls die Anzahl der Daten zu groß ist, um damit in einer sinnvollen Zeit Analysen durchführen zu können. Dabei kann man zwischen einer zufälligen Stichprobe und einer geschichteten (stratified) Stichprobe wählen.

Bei „Transform“ kann man verschiedene Transformationen auf den Daten anwenden. Dabei ist abhängig vom Datentyp nur eine gewisse Anzahl an Transformationen möglich. Für Datumsattribute sind nur zwei Transformationen verfügbar:

- Diskretisierung und
- Behandlung von fehlenden Werten.

Zusätzlich dazu ist bei kategorischen Attributen noch eine benutzerdefinierte Transformation möglich.

Für Numerische Attribute kann man dazu noch folgende Transformationen durchführen:

- Normalisierung und
- Ausreißerbehandlung.

Für die Normalisierung existieren einige Methoden im Oracle Data Miner.

3.4 Oracle Data Mining Funktionen / Algorithmen

In diesem Abschnitt werden die angebotenen Funktionen und die dazugehörigen Algorithmen dargestellt.

Wie bereits im Punkt Grafische Benutzeroberfläche der Unterkapitels 3.2 Beschreibung des Oracle Data Miners beschrieben werden die Problemtypen des Oracle Data Miners im Bereich „Models“ der „Component Palette“ im „Oracle Data Miner“ ausgewählt.

Hier soll nun eine Übersicht der im Oracle Data Miner implementierten Problemtypen aufgezeigt werden. Dabei soll nur auf die Funktionen die für das Data Mining wichtig sind geschaut werden.

Problemtyp	Algorithmus
Klassifikation	Support Vector Machine
	Entscheidungsbaum
	Verallgemeinerte Lineare Modelle (Logistische Regression)
	Naive Bayes
Prognose / Regression	Support Vector Machine
	Verallgemeinerte Lineare Modelle (Lineare Regression)
Clustering	k-Means
	O-Cluster
Assoziationsanalyse	Apriori
Abweichungsanalyse	Ein – Klassen – Support Vector Machine

Tabelle 1: Übersicht Problemtypen mit Algorithmen

Abweichungsanalyse

Für die Abweichungsanalyse im Oracle Data Miner benötigt man Daten die bereits einer Klasse zugeordnet wurden. Die Abweichungsanalyse soll dann anhand der Rate der Ausreißer bestimmen wie viele Daten als Ausreißer deklariert werden. Dabei muss man allerdings beachten, dass auch Fälle als Ausreißer klassifiziert werden können die in Wirklichkeit gar keine sind. Das liegt daran, dass immer ein gewisser Anteil an Ausreißern bestimmt wird. Deshalb ist es ratsam diese identifizierten Fälle

von einem Experten auf diesem Gebiet untersuchen zu lassen. Der Standardwert für die Rate der Ausreißer liegt bei 10 %.

Assoziationsanalyse

Bei der Assoziationsanalyse im Oracle Data Miner wird vorrangig auf die Warenkorbanalyse Wert gelegt. Dabei muss man eine Transaktion und das Attribut mit dem gekauften Artikeln vorgeben. Man kann auch die Anzahl oder Typ des Artikels für die Analyse unter „Item Value“ angeben. Dabei muss dieser Wert weniger als 10 verschiedene Werte besitzen. Der Standard dafür ist „Existence“.

Für den Apriori Algorithmus im Oracle Data Mining kann man die maximale Länge der Regel, die minimale Konfidenz und den minimalen Support einstellen.

Klassifikation

Die Klassifikation im Oracle Data Miner benötigt eine Zielvariable. Diese enthält die bereits vorgegebene Klassifikation für die Fälle.

Die Entscheidungsbaumerstellung läuft nach einem binären Split ab. Als Homogenitätsmetrik kann man zwischen Entropy und GINI wählen. Man hat dazu noch weitere Einstellungsmöglichkeiten. Wie z. B. die kleinste Anzahl an Einträgen ab dem das Splitting der Knoten durchgeführt wird. Oder die minimale Anzahl an Einträgen in einem Knoten, sodass keine zu kleinen Knoten entstehen.

Der Vorteil bei Entscheidungsbäumen liegt in der Interpretierbarkeit der entstandenen Regeln. Durch den einfachen Aufbau ist die benötigte Zeit beim Erstellen eines Entscheidungsbaums vergleichsweise gering. Der

Entscheidungsbaum benötigt intern keinerlei Datenaufbereitung. Diese ist intern im Algorithmus verankert.

Einige der Vorteile zeigen einen direkten Weg zu den Nachteilen eines Entscheidungsbaumes. Aufgrund des einfachen Aufbaus ist die Genauigkeit bei problematischen Klassengrenzen schlechter als bei Algorithmen, die eine flexiblere Klassifikation zulassen.

Die Support Vector Machine läuft im Oracle Data Mining mit numerischen Variablen. Kategorische Variablen werden automatisch in binäre numerische Variablen transformiert. Es werden zwei Kernelfunktionen, linear und gaußsch, unterstützt. Mit Hilfe von Active Learning werden die Daten automatisch nur in eine interessante Richtung hin untersucht. Man kann auch den Komplexitätsfaktor, der Overfit verhindern soll, manuell anpassen. Man sollte diesen Wert jedoch nicht zu groß wählen, da das Modell dann einfach zu komplex für allgemeinere, nicht in den Trainingsdaten vorhandene, Fälle wäre.

Der Vorteil der Klassifikation mit Support Vector Machines ist, dass die erstellten Regeln in der Regel ziemlich gute Testergebnisse aufweisen. Der Nachteil dieser Methode ist, dass die Erstellung des Klassifikators sehr viel Zeit in Anspruch nehmen kann und man die Regeln, die der Algorithmus erstellt nicht vom Menschen verstanden werden können.

Bei „Naive Bayes“ muss man besonders darauf achten, dass man keine abhängigen Variablen zur Klassifikation benutzt. Man kann dafür nur die Schranken für die paarweisen und einzelnen Wahrscheinlichkeiten angeben ab denen diese Wahrscheinlichkeiten im Endergebnis berücksichtigt werden.

Bei den verallgemeinerten linearen Modellen wird die logistische Regression verwendet. Dafür kann man Gewichte für die Attribute festlegen falls man den Daten nicht traut. Dieser Algorithmus funktioniert nur bei einer binären Klassifikation.

Clustering

Für das Clustering sind zwei Algorithmen verfügbar, der k-Means und OCluster. Im Oracle Data Mining sind beide als hierarchische und probabilistische Verfahren implementiert.

Für k-Means muss dazu die Anzahl k der Cluster vorgegeben werden. Als Distanzmetrik kann zwischen der euklidischen Metrik und dem Kosinus gewählt werden. Der Kosinus stellt dabei den Winkel zwischen den Attributen dar. Ferner kann noch die Anzahl der Iterationen festgelegt werden.

O-Cluster ist im Gegensatz dazu ein dichtebasierter Algorithmus. Dabei werden „Täler“ zwischen zwei dichten „Bergen“ gesucht und anhand dieser Entscheidungsebene die Cluster gebildet.

Regression

Mit Hilfe der Regression kann man beim Oracle Data Mining die Prognose, die bereits im Unterkapitel 2.5 Problemtypen auf Seite 19 beschrieben wurde, bearbeiten. Für die Regression sind 2 Algorithmen verfügbar, die Support Vector Machine und die verallgemeinerten linearen Modelle. Bei den verallgemeinerten linearen Modellen wird dazu die lineare Regression benutzt und man geht davon aus, dass die Varianz über dem kompletten Wertebereich des Attributs gleich ist.

4 Fallstudie „InEK – Daten“

In diesem Kapitel werden zunächst die InEK-Daten (Institut für das Entgeltsystem im Krankenhaus) am Anfang beschrieben, sodass man einen Überblick erhält, auf welcher Datenbasis das Data Mining durchgeführt werden soll. Der Data-Mining-Untersuchung lagen einige Fragestellungen zugrunde, die von Mitarbeitern des SMS (Sächsisches Staatsministeriums für Soziales und Verbraucherschutz) zur Verfügung gestellt wurden. Diese wurden daraufhin untersucht inwiefern Data Mining Methoden darauf angewendet werden können, um zufriedenstellende Antworten liefern. Die Bereitstellung von Aufgaben ist sehr nützlich und notwendig, da man ansonsten manuell nach Mustern suchen müsste. Das Problem bei der Suche ohne konkrete Fragestellungen ist, dass im Allgemeinen sehr viele Analysemethoden, Attribute und Fälle vorhanden sind. Damit wird die Anzahl der möglichen Muster sehr groß. Aber nur ein kleiner Teil von diesen Mustern ist im Allgemeinen nützlich (siehe: Definition KDD auf Seite 4 im Unterkapitel 2.1 Einführung – Data Mining). Um die Nützlichkeit messen zu können, benötigt man eine Möglichkeit des Vergleichs einer Lösung mit einer Fragestellung, inwiefern diese sinnvoll beantwortet wurde.

Die Durchführung des Data Minings auf den InEK-Daten erfolgte anhand eines bereits bewährten Ablaufplans, da man somit einen besseren Überblick über den Data Mining Prozess erhält. Dieser Ablaufplan wird durch CRISP-DM, das bereits im Punkt 2.1 Einführung – Data Mining auf Seite 6 beschrieben wurde, dargestellt.

Die Datenanalyse wird mittels des Oracle Data Miners beschrieben. Diese Vorgehensweise wurde bereits im Unterkapitel 3.2 Beschreibung des Oracle Data Miners detailliert beschrieben.

4.1 Datenbasis

Für das Data Mining mit den InEK-Daten liegen einige Tabellen in einem Data-Warehouse im Sternschema vor. Dabei gibt es eine Reihe von Dimensionstabellen und einige Faktentabellen. Diese sollen im Folgenden beschrieben werden.

Die folgenden Faktentabellen sind verfügbar:

- FAKT_ICD (International Statistical Classification of Diseases and Related Health Problems)
- FAKT_OPS (Operationen- und Prozedurenschlüssel)
- FAKT_FAELLE
- FAKT_FAB (Fachabteilung)
- FAKT_CMI (Case Mix Index)
- KHH (Krankenhaushäufigkeit) Tabellen
    - FAKT_KHH_ICD
    - FAKT_KHH_OPS
    - FAKT_KHH_FAB

Dazu gehören noch 20 Dimensionstabellen, die weiterführende Informationen zu den in den Faktentabellen enthaltenen Einträgen liefern.

Im weiteren Verlauf der Arbeit werden alle Faktentabellen bis auf FAKT_CMI, FAKT_KHH_OPS, FAKT_KHH_FAB und FAKT_KHH_ICD näher dargestellt. Eine Aufzählung über alle Faktentabellen und Dimensionstabellen findet man im Betriebshandbuch (Robotron Datenbank-Software

GmbH, 2010a) und in der Dokumentation (Robotron Datenbank-Software GmbH, 2010b) für die SMS-ZFD ().

In Abbildung 6 werden die Verbindungen zwischen den einzelnen Dimensions- und Faktentabellen dargestellt.

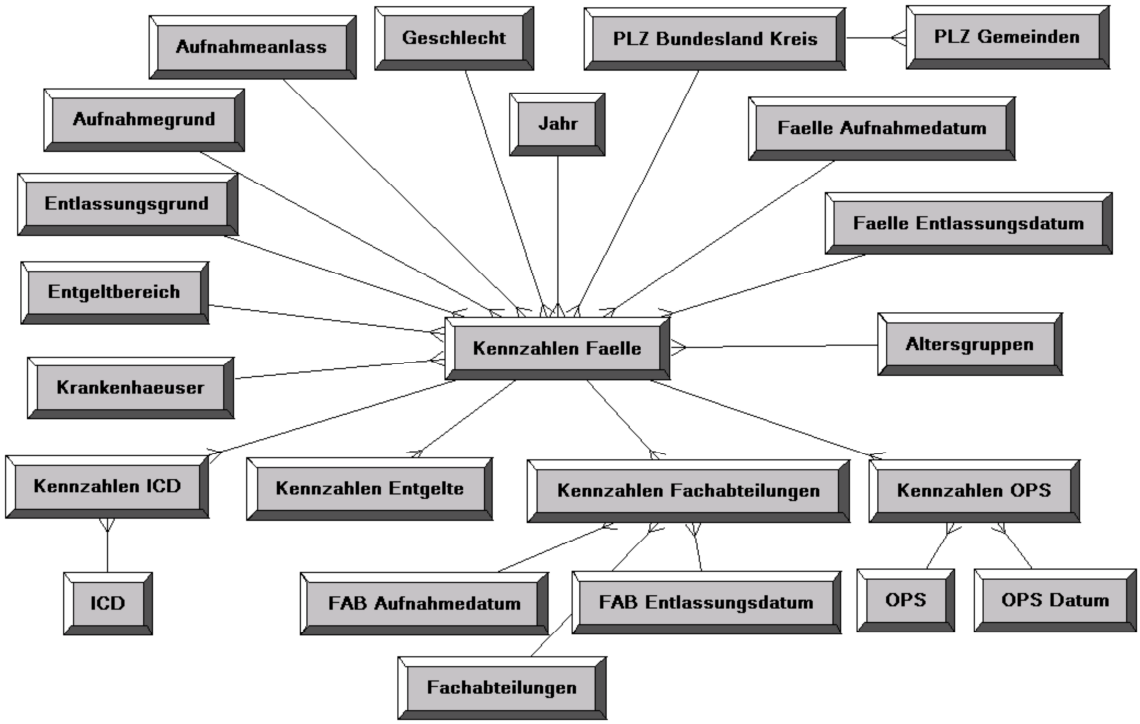

Abbildung 6: Sternschema für die InEK-Daten (Robotron Datenbank-Software GmbH, 2010a)

Die Faktentabelle FAKT_FAELLE enthält alle Falldaten von 6.128.823 Fällen, die in einem sächsischen Krankenhaus in den Jahren 2004-2009 entlassen wurden. Die Eintragung eines Falles in die Tabelle geschieht dabei im Jahr der Entlassung.

Zu den Falldaten gehören u. a. Angaben zum Krankenhaus indem der Fall aufgenommen wurde, demographische Daten über den Patienten und einige Angaben über die Einordnung des Falles. Zur Einordnung des Falles kann man u. a. das Jahr der Datenerhebung, Datum der Aufnahme / Entlassung, Aufnahmeanlass, Aufnahmegrund, Entlassungsgrund und den Entgeltbereich angeben.

Zu den Fällen existieren in der Faktentabelle FAKT_ICD alle Diagnosen die im Laufe des Aufenthalts im Krankenhaus getroffen wurden. Insgesamt handelt es sich dabei um 31.222.690 Einträge. Dabei gibt es eine Hauptdiagnose HD mit einer oder mehrerer Sekundär- / Nebendiagnosen (SD /ND) für einen bestimmten Fall. Dementsprechend beinhaltet diese Tabelle eine oder mehrere Einträge für jeden Fall aus der Faktentabelle Fälle. Die getroffene Diagnose kann man über das Attribut ICD_Kode einordnen. Die dazugehörige Dimensionstabelle DIM_ICD beinhaltet alle Diagnosen die getroffen werden können.

Die Fachabteilungen in denen sich ein Fall befindet werden in der Faktentabelle FAKT_FAB dargelegt. Dabei handelt es sich insgesamt um 7.232.951 Einträge. Dabei werden Angaben über die Abteilungsart, die Art der FAB, das Datum der Aufnahme und das Datum der Entlassung aufgezeichnet. Diese Tabelle ist mit einer Dimensionstabelle DIM_FAB, mit einer Auflistung aller mögl. Fachabteilungen, verbunden.

In der Faktentabelle FAKT_OPS werden alle zu den Fällen durchgeführten Maßnahmen, Operationen und Prozeduren aufgelistet. Dabei handelt es sich um 14.988.770 Einträge. Dabei werden u. a. Attribute zur Beschreibung und zum Datum der Ausführung des OPS gespeichert. Diese Tabelle ist mit der Dimensionstabelle DIM_OPS, mit einer Auflistung aller mögl. OPS Kodes, verbunden.

Die Faktentabelle FAKT_ENTGELTE beinhaltet die verschiedenen Entgeltarten die während der Aufenthaltsdauer des Falles angefallen sind. Insgesamt sind 44.039.637 Einträge vorhanden. Diese Tabelle kann noch mit den Dimensionstabellen DIM_DRG_BWR_HA und DIM_DRG_BWR_BA verbunden werden. Diese enthalten Informationen über die DRG-Fallpauschalen.

Eine Beschreibung der meisten Attribute ist teilweise in den Kommentaren für die Spalten in den einzelnen Tabellen vorhanden. Des Weiteren kann man sich die Bedeutungen der einzelnen Attribute an die Dokumentation der InEK Daten (InEK 1, 2, 3) wenden.

4.2 Fragestellungen

Für das Data Mining wurden von Mitarbeitern des SMS einige Fragestellungen vorgeschlagen mit denen überprüft werden sollte ob diese mit den Algorithmen in Oracle Data Mining beantwortbar sind. Es handelt sich hierbei um die folgenden Fragestellungen:

Fragestellung 1

Eine Untersuchung im Rahmen der septischen Chirurgie, d.h. bestimmte ICD als Haupt- oder Nebendiagnose zusammen mit ausgewählten OPS – und das Ganze als Häufung nach Krankenhäusern.

Fragestellung 2

Ein Vergleich des Entlassungsdatums in einem Krankenhaus mit dem Aufnahmedatum in einem anderen Krankenhaus bei gleichen Patienteninitialen (Alter, Geschlecht, PLZ) – ggf. auch unter Berücksichtigung der Hauptdiagnose.

Fragestellung 3

Die InEK-Daten sollen darauf hin untersucht werden, wie sich in den einzelnen Krankenhäusern im Freistaat Sachsen und in ausgewählten Fachgebieten Neurochirurgie, Neurologie, Innere Medizin und Chirurgie das Leistungsspektrum (DRG und/oder OPS) bei Hauptdiagnose C 71 Gehirnkrebs über die Jahre entwickelt hat.

4.3 Bearbeitung der Fragestellungen

In diesem Unterkapitel soll die Vorgehensweise für das Data Mining anhand der Fragestellungen dargestellt werden. Im Laufe des Bearbeitens hat sich herausgestellt, dass sich bestimmte Aufgaben besser für die Analyse mit Data-Mining-Techniken eignen als andere.

In den folgenden Detailbetrachtungen wurden die Punkte des CRISP-DM gesondert für jede Fragestellung abgearbeitet, da die Aufgaben in der Regel unterschiedliche Methoden und eine andere Herangehensweise erfordern. Das Deployment wurde bei allen Fragestellungen weggelassen.

Oracle Data Miner

Die Aufgaben wurden mit dem Oracle Data Miner bearbeitet. Für jede Aufgabenstellung wurde im Oracle Data Miner ein eigenes Projekt erstellt. Für die einzelnen Unteraufgaben wurden Workflows in den jeweiligen Projekten erstellt. Die Vorgehensweise dafür wurde im Kapitel 3 dargelegt.

4.3.1 Fragestellung 1

Business Understanding

Zum „Business Understanding“ gehört, dass man die Probleme / Aufgabe, für die das Data Mining angewendet werden soll, versteht. Das wird durch die Bereitstellung der folgenden Aufgabenstellung erledigt.

Eine Untersuchung im Rahmen der septischen Chirurgie, d.h. bestimmte ICD als Haupt- oder Nebendiagnose zusammen mit ausgewählten OPS – und das Ganze als Häufung nach Krankenhäusern.

Diese Aufgabe beinhaltete mehrere Teilaufgaben:

1. alle KH-Fälle mit kodierter Nebendiagnose U80* und U81* – Häufungen nach Krankenhäusern (ICD – Diagnoseart = ND)
2. wie 1. nur mit den OPS: 5-821* (Revision, Wechsel und Entfernung einer Endoprothese am Hüftgelenk) bzw. 5-823* (Revision, Wechsel und Entfernung einer Endoprothese am Knie)
3. wie 1. nur mit OPS 8-987 und T84.5*, T84.6* u. T84.7* als Hauptdiagnose (ICD – Diagnoseart = HD)

Bei dieser Aufgabe stellte sich das Problem, dass diese eigentlich nicht für das Data Mining in der gestellten Form sinnvoll ist. Es ist dabei eher sinnvoll diese Aufgabe mittels anderer Methoden, wie z. B. einigen SQL-Befehlen und einer Darstellung der Ergebnisse in Excel, durchzuführen.

Nichtdestotrotz wurde versucht ein Data Mining Modell darauf anzuwenden. Dabei muss vorerst entschieden werden, welcher Data Mining Problemtyp angewendet werden sollte. Dabei wurde eine Assoziationsanalyse oder eine Klassifikation zur Analyse in Erwägung gezogen. Die Vorgehensweise und Machbarkeit soll in den weiteren Punkten dargestellt werden.

Bei einer Klassifikation soll versucht werden die Fälle, die untersucht werden sollten, im Vergleich zu den restlichen Fällen zu klassifizieren. Somit könnte man anhand einer Entscheidungsregel erkennen was die großen Unterschiede zwischen den beiden Klassen ausmacht.

Bei der Assoziationsanalyse werden Auffälligkeiten in den Daten zwischen dem Krankenhaus und dem Jahr des Auftretens des Falles gefunden.

Data Understanding

Beim Punkt „Data Understanding“ werden alle Daten die zur Bearbeitung der Aufgabe nötig sind beschafft. Des Weiteren werden die zur Verfügung stehenden Daten vorgestellt. Für diese Aufgabe werden die Faktentabellen FAKT_ICD, FAKT_OPS und FAKT_FAELLE benötigt. Für die einzelnen Teilaufgaben müssen einige SQL-Befehle ausgeführt werden, um die Fälle mit den gewünschten Eigenschaften zu erhalten.

Der Befehl:

```
create or replace force view "KH"."1_AUFGABE_1_1"
as
  select * from fakt_faelle where id in (
    select fafall_id from fakt_icd where
    icd_kode like 'U80%' or icd_kode like 'U81%');
```

wählt alle Fälle aus, die den Eigenschaften aus der ersten Teilaufgabe entsprechen. Das kann man danach auch für die anderen Teilaufgaben durchführen.

Um die Häufungen bzgl. der Krankenhäuser in einem Jahr zu erhalten kann man mit einem SQL-Befehl mit Hilfe einer group-by Klausel arbeiten:

```
select ik, jahr_datenerhebung , count(*) anz from
"1_AUFGABE_1_1" group by ik, jahr_datenerhebung;
```

Diese Daten kann man danach mit Hilfe von Excel in einem Diagramm darstellen. Dabei wurden nicht nur die absoluten, sondern auch die relativen Anzahlen dargestellt.

Bei der Bearbeitung der Data Mining Aufgabe bzgl. der Klassifikation stellt sich das Problem, dass teilweise sehr wenige Daten zur Verfügung stehen und somit eine Klassifikation keinen erkennbaren Sinn ergibt.

Data Preparation

Für diese Aufgabenstellung müssen die Daten für die Assoziationsanalyse vorbereitet werden. Dafür müssen die aus der „Data Understanding“ bereitgestellten Daten in eine Form gebracht werden, die mit dem Oracle Data Miner bearbeitet werden kann. Dabei muss man beachten, dass die Assoziationsanalyse normalerweise im Oracle Data Miner für die Warenkorbanalyse ausgelegt ist. Man kann die Daten mittels einer SQL-Anweisung in die gewünschte Form bringen.

Die SQL-Anweisung für die Erstellung der Assoziationsanalyse lautet wie folgt:

```
CREATE OR REPLACE FORCE VIEW "KH"."1_INIT_ASSOZ"
("ID", "LN", "KHAUS_JAHR")
AS
  SELECT f.id, 1 ln,
         f.ik||d.kh_name khaus_jahr
  FROM fakt_faelle f, dim_krankenhaeuser d
  WHERE f.id IN
    (SELECT fafall_id
    FROM fakt_icd WHERE icd_kode LIKE 'U80%'
                     OR icd_kode LIKE 'U81%')
  AND d.id = f.dimkh_id
  UNION
  SELECT id,
    2 ln,
    TO_CHAR(jahr_datenerhebung) khaus_jahr
  FROM fakt_faelle
  WHERE id IN
    (SELECT fafall_id
    FROM fakt_icd
    WHERE icd_kode LIKE 'U80%'
    OR icd_kode LIKE 'U81%'
    );
```

Dadurch kann man Auffälligkeiten zwischen dem Krankenhaus und dem Jahr in dem der Fall aufgetreten ist finden.

Modeling

Nachdem die View "1_INIT_ASSOZ" im Punkt „Data Preparation“ erstellt worden ist, kann man diese zur Assoziationsanalyse benutzen. Dabei wird ein Workflow (siehe Abbildung 7) erstellt bei dem man mit Hilfe des Apriori Algorithmus eine Assoziationsanalyse durchführen kann. Dabei wurden für den minimalen Support und die minimale Konfidenz 0 eingegeben um alle möglichen Assoziationsregeln zu finden. In Tabelle 2 kann man einen Auszug aus den erzeugten Assoziationsregeln sehen. Diese Assoziationsregeln kann man wie folgt interpretieren. Falls ein Krankenhaus in der Bedingung ist, so bedeutet das, dass Support% aller Infektionen in diesem Krankenhaus und in dem Jahr, das in der Folgerung vorhanden ist, aufgetreten sind. Für die Konfidenz bedeutet das, dass Konfidenz% aller Infektionsfälle in diesem Krankenhaus in dem Jahr aufgetreten sind. Falls das Jahr in der Bedingung auftritt, kann man die Konfidenz als den Prozentsatz ansehen mit dem das Krankenhaus, als Folgerung der Regel, an Infektionen in diesem Jahr beteiligt war.

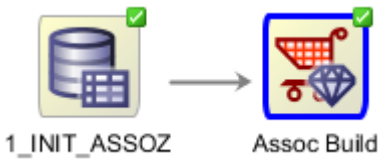

Abbildung 7: Workflow zur Fragestellung 1

Bedingung	Folgerung	Konfidenz	Support
Bavaria-Klinik Kreischa	2009	100	1,0987
Diakonissenkrankenhaus Dresden	2009	56,1939	1,0549
Oberlausitz-Kliniken gGmbH Krankenhaus Bischofswerda	2009	54,4118	0,6235
St. Elisabeth-Krankenhaus Leipzig gGmbH	2009	52,9412	0,4246
HELIOS Vogtland-Klinikum Plauen GmbH	2009	50,7407	0,9234
Neurologisches Rehabilitationszentrum Leipzig	2009	50,5415	0,4718
Vogtlandklinikum Plauen GmbH	2006	47,1591	0,2797

Klinikum des Landkreises Löbau-Zittau gGmbH	2009	46,1741	0,5898
Oberlausitz-Kliniken gGmbH Krankenhaus Bautzen	2009	45,641	1,1998
Krankenhaus Weisswasser gGmbH	2009	45,3659	0,3134
Kreiskrankenhaus Delitzsch GmbH - Klinik Delitzsch	2009	45,1852	0,2056
HELIOS Klinikum Aue	2009	43,9114	1,2032

Tabelle 2: Assoziationsregeln zur Fragestellung 1

Im Folgenden werden die Ergebnisse der Analyse für die Anwendung einer einfachen group-by Klausel dargestellt.

In Tabelle 3 kann man die Anzahl der absoluten Krankenhausinfektionen für die Jahre sehen. Man sieht, dass die Infektionen im Klinikum Chemnitz gGmbH ziemlich stark in den letzten Jahren angestiegen sind. Warum das so ist kann man höchstens durch einen starken Anstieg an Fallzahlen erklären.

Jahr	Krankenhaus	Anzahl
2009	Klinikum Chemnitz gGmbH	1284
2008	Klinikum Chemnitz gGmbH	848
2008	Universitätsklinikum Leipzig, AöR	595
2007	Universitätsklinikum Leipzig, AöR	560
2007	Klinikum Chemnitz gGmbH	540
2009	Universitätsklinikum Leipzig, AöR	520
2009	Universitätsklinikum Carl Gustav Carus an der TU Dresden, AöR	474
2009	HELIOS Kliniken Leipziger Land	447
2008	Universitätsklinikum Carl Gustav Carus an der TU Dresden, AöR	445
2006	Universitätsklinikum Leipzig, AöR	382
2009	HELIOS Klinikum Aue	357
2009	Klinikum St. Georg gGmbH Leipzig	357
2009	Oberlausitz-Kliniken gGmbH Krankenhaus Bautzen	356
2009	Krankenhaus Dresden-Friedrichstadt – Städtisches Klinikum	351
2005	Universitätsklinikum Leipzig, AöR	327
2009	Bavaria-Klinik Kreischa	326
2007	Klinikum Hoyerswerda gGmbH	323

Tabelle 3: Anzahl der Krankenhausinfektionen absolut

Man kann, falls man die Fallzahlen außer Acht lassen möchte auch die relativen Krankenhausinfektionen betrachten. Dabei wird die absolute Anzahl durch die Anzahl aller Fälle in diesem Krankenhaus in diesem

Jahr geteilt. In Tabelle 4 kann man die Ergebnisse anschauen. Dabei stellt man fest, dass jetzt das Neurologische Rehabilitationszentrum Leipzig und die Bavaria-Klinik Kreischa stark vertreten sind.

Jahr	Anteil gesamt in %	Krankenhaus
2009	26,3653484	Neurologisches Rehabilitationszentrum Leipzig
2009	25,99681021	Bavaria-Klinik Kreischa
2008	21,01449275	Neurologisches Rehabilitationszentrum Leipzig
2007	15,15151515	Neurologisches Rehabilitationszentrum Leipzig
2009	2,394512037	Oberlausitz-Kliniken gGmbH Krankenhaus Bischofswerda
2009	2,28717574	Diakonissenkrankenhaus Dresden
2009	1,858859733	HELIOS Kliniken Leipziger Land
2009	1,666432617	Oberlausitz-Kliniken gGmbH Krankenhaus Bautzen
2009	1,635033158	HELIOS Krankenhaus Leisnig
2009	1,597312932	Klinikum Chemnitz gGmbH
2008	1,556944955	HELIOS Krankenhaus Leisnig
2005	1,490346618	Park-Krankenhaus Leipzig-Südost GmbH
2006	1,47409945	Park-Krankenhaus Leipzig-Südost GmbH
2009	1,454760497	HELIOS Klinik Schkeuditz
2007	1,417517394	Weißeritztal-Kliniken GmbH

Tabelle 4: Anteil der Krankenhausinfektionen pro Jahr und Krankenhaus

Evaluation

Die Assoziationsregeln für diese Aufgabe sind durchaus nützlich, falls zu viele Krankenhäuser analysiert werden sollen. Damit kann man evtl. herausfinden welche Krankenhäuser besonders in einem Jahr von Infektionen betroffen sind und wo die meisten Infektionen auftreten. Trotzdem ist es in vielen Fällen besser keine Data Mining Algorithmen auf für das Data Mining zu einfache Fragestellungen anzuwenden.

4.3.2 Fragestellung 2

Business Understanding

Die Fragestellung lautet wie folgt:

Ein Vergleich des Entlassungsdatums in einem Krankenhaus mit dem Aufnahmedatum in einem anderen Krankenhaus bei gleichen Patienteninitialen (Alter, Geschlecht, PLZ) – ggf. auch unter Berücksichtigung der Hauptdiagnose.

Für diese Aufgabe werden jeweils 2 Fälle aus der Faktentabelle FAKT_FAELLE anhand gewisser Kriterien verglichen. Dafür eignet sich die Assoziationsanalyse sehr gut, da man damit nicht nur einzelne sondern auch mehrere Fälle vergleichen kann. Dadurch kann man dann u. U. Häufungen bei Aufnahme- und Entlassungskrankenhäusern feststellen und somit erkennen zwischen welchen Krankenhäusern besonders oft Fälle verlegt werden.

Data Understanding

Beim Data Understanding werden die Daten die zur Data Mining Analyse benötigt werden beschafft und ausgewählt. Für diese Aufgabenstellung werden die Faktentabelle FAKT_FAELLE und FAKT_ICD benötigt. Ebenso ist es wünschenswert die Dimensionstabelle DIM_KRANKENHAEUSER zur Analyse zu benutzen. Damit kann man die Orte und Namen der Krankenhäuser anhand des IK (Institutskennzeichen) identifizieren. Dabei muss man allerdings beachten, dass bei manchen Krankenhäusern nur das IK vorhanden ist und keine weiteren Angaben vorliegen. Diese fehlenden Daten kann man sich mit Hilfe des IK aber leicht mit Hilfe einer Suche im Internet verschaffen.

Bei dieser Aufgabenstellung müssen jeweils 2 Fälle miteinander verglichen werden. Dabei bezeichnet der eine Fall eine Entlassung aus einem Krankenhaus, wobei der andere Fall eine Aufnahme in einem Krankenhaus bezeichnet.

Mit Hilfe dieses Workflows kann man die Vorbereitung der Daten und die Erstellung des Views „2_AE_INITIALEN_HD“ nachvollziehen:

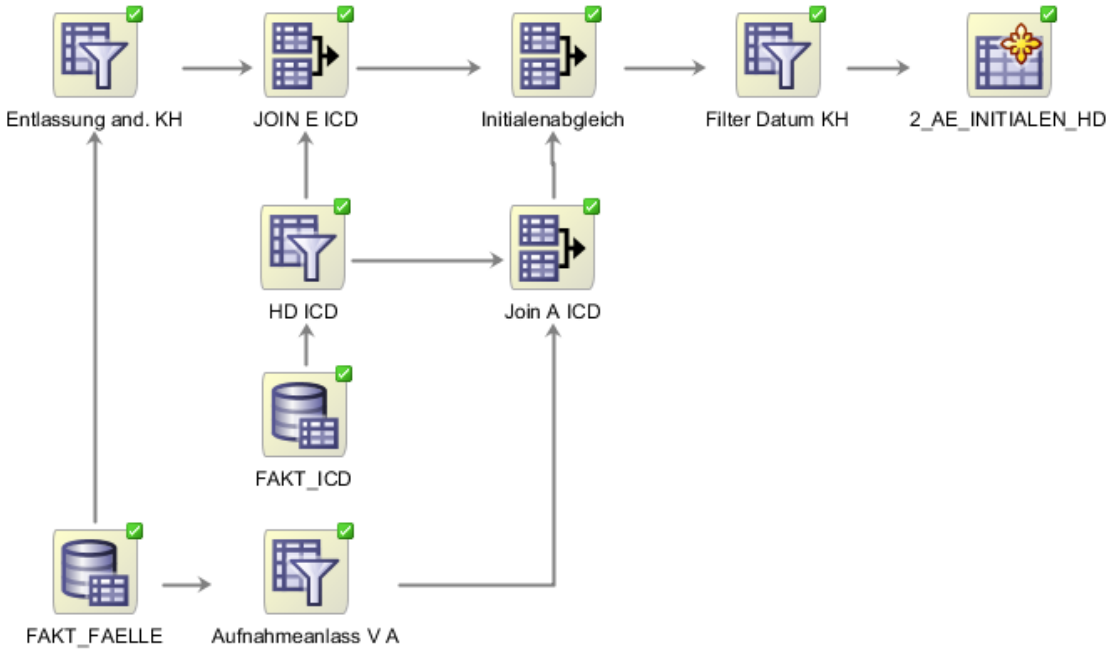

Abbildung 8: Workflow zum Erzeugen von „2_AE_INITIALEN_HD“

In „Entlassung and. KH“ bzw. „Aufnahmeanlass V A“ werden die Entlassungs- und Aufnahmefälle ausgewählt. Bei der Entlassung aus einem Krankenhaus werden alle Fälle gemäß der folgenden Regel betrachtet:

ENTLASSUNGSGRUND >= 60 and ENTLASSUNGSGRUND < 70

Bei der Aufnahme in ein Krankenhaus werden alle Fälle bei denen das Attribut AUFNAHMEANLASS V oder A ist ausgewählt.

Danach werden für die Aufnahme- und Entlassungstabellen die Hauptdiagnosen der jeweiligen Fälle mit einbezogen.

Beim Initialenabgleich werden die Ausnahme- und Entlassungsfälle miteinander verbunden. Dabei werden die Tabellen über Inner Joins anhand des Alters, der PLZ, des Geschlechts und der Hauptdiagnose miteinander verbunden. Somit sind in der erzeugten Tabelle nur noch Fälle vorhan-

den, bei denen die Initialen für die Entlassung und die Aufnahme gleich sind. Dadurch stellt man allerdings nicht sicher, dass es sich um dieselbe Person handelt.

Da man mit Hilfe eines Joins im Oracle Data Miner keine weiteren Einschränkungen bzgl. der Ignorierung von bestimmten Fällen vornehmen kann, muss man einen weiteren Knoten anbringen um das zu bewerkstelligen. Das wurde hier mittels des Knotens „Filter Datum KH“ bewerkstelligt. Dabei werden nur Fälle betrachtet bei denen die Entlassung und die Aufnahme maximal vier Tage auseinander liegen. Ebenso werden Fälle ignoriert bei denen das Aufnahme- / Entlassungskrankenhaus gleich ist.

Data Preparation

Bei der „Data Preparation“ werden die zur Verfügung stehenden Daten so vorbereitet, dass ein Data Mining Modell im „Modeling“ darauf angewendet werden kann.

Mit Hilfe des Views „2_AE_INITIALEN_HD“ kann man danach, für die Assoziationsanalyse den View „2_AE_INIT_HD_ASSOZ“ erstellen.

```
CREATE OR REPLACE FORCE VIEW "KH"."2_AE_INIT_HD_ASSOZ"
("ID", "LN", "KH")
AS
  SELECT f.id||' '||f.id_a,
    1 ln, f.ik||'Entlassungskrankenhaus'||d.kh_name kh
  FROM "2_AE_INITIALEN_HD" f,
    dim_krankenhaeuser d
  WHERE d.id = f.dimkh_id
  UNION
  SELECT f.id||' '||f.id_a,
    2 ln, f.ik_a||'Aufnahmekrankenhaus'||d.kh_name kh
  FROM "2_AE_INITIALEN_HD" f,
    dim_krankenhaeuser d
  WHERE d.id = f.dimkh_id_a;
```

Modeling

Nachdem die View „2_AE_INIT_HD_ASSOZ“ beim Punkt „Data Preparation“ erstellt worden ist, kann man eine Assoziationsanalyse mit dem View „2_AE_INIT_HD_ASSOZ“ durchführen. Man kann ebenfalls die Hauptdiagnose ignorieren um somit nur die Initialen als Abgleich zu benutzen.

Dafür wird ein Workflow (siehe Abbildung 9) erstellt um damit die Analyse durchführen zu können. Der Oracle Data Miner verwendet dafür den Apriori-Algorithmus. Dabei kann man die maximale Länge der Regeln, die minimale Konfidenz und den minimalen Support auswählen. Dabei wurde, um alle möglichen Regeln zu erstellen die maximale Länge der Regeln auf das Maximum von 20 gesetzt. Die minimale Konfidenz und der minimale Support wurden analog dazu auf das Minimum von 0% gesetzt. Nach dem Ausführen der Assoziationsanalyse wurden insgesamt für die Variante mit der Hauptdiagnose 2218 Regeln erzeugt. Man sollte allerdings, um die Anzahl der Regeln zu verringern und um unwichtige Regeln auszuschließen, sie gemäß einer unteren Schranke für die Konfidenz, den Support oder den Lift filtern. Dadurch kann man Regeln ausschließen, die nur einen Support von wenigen Fällen besitzen. Bei dieser Aufgabe zeigt der Support an wie viele Aufnahmen / Entlassungen es zwischen den beiden Krankenhäusern die in der Regel aufgelistet werden gibt. Die Konfidenz zeigt im Gegensatz dazu an, welchen Anteil eine Entlassung in ein Krankenhaus oder eine Aufnahme aus einem Krankenhaus im Verhältnis mit allen Entlassungen aus diesem Krankenhaus oder Aufnahmen in dieses Krankenhaus besitzt. Der Lift sagt etwas über die Bedeutung der Regel aus. Dabei muss man allerdings beachten, dass bei einem zu kleinen Support der Lift teilweise zu groß wird und somit die Bedeutung einer Regel überbewertet wird.

In dieser Fragestellung zeigt ein hoher Lift u. a. an, ob ein Krankenhaus viele Fälle von einem anderen Krankenhaus aufnimmt ohne dabei viel von anderen Krankenhäusern aufzunehmen. Das gilt analog mit entlassenden Krankenhäusern. Darüber hinaus ist der Lift aber auch groß wenn der Support der Bedingung, also die Anzahl der Fälle in einem Krankenhaus, relativ klein ist. Das alles kann man an der Formel zum Lift leicht ablesen.

Als Beispiel kann man folgende Regel angeben:

Aufnahmekrankenhaus: Neurologisches Rehabilitationszentrum Leipzig
Entlassungskrankenhaus: Universitätsklinikum Leipzig, AöR
Lift: 35.6454
Konfidenz: 100%
Support: 0.0803%

Dabei zeigt sich, dass 100% aller Aufnahmen in das Neurologisches Rehabilitationszentrum Leipzig aus dem Universitätsklinikum Leipzig, AöR entlassen wurden. In 0.0803% aller Fälle wurde ein Fall aus dem Universitätsklinikum Leipzig, AöR in das Neurologisches Rehabilitationszentrum Leipzig verlegt.

Es wurden Assoziationsregeln für Verlegungen mit gleicher Hauptdiagnose und ohne gleiche Hauptdiagnose durchgeführt. Einen Teil der Ergebnisse kann man in Tabelle 5 und Tabelle 6 sehen.

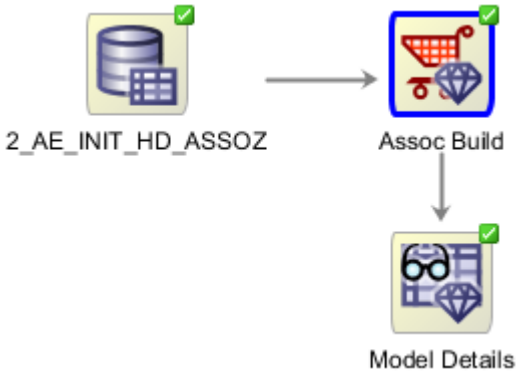

Abbildung 9: Workflow zum Data-Mining Modell von Fragestellung 2

Bedingung	Folgerung	Lift	Konfidenz	Support
Aufnahme: Kreiskrankenhaus Delitzsch GmbH – Klinik Delitzsch	Entlassung: Kreiskrankenhaus Delitzsch GmbH – Klinik Eilenburg	5,3	71,4	0,3
Entlassung: Kreiskrankenhaus Delitzsch GmbH – Klinik Eilenburg	Aufnahme: Kreiskrankenhaus Delitzsch GmbH – Klinik Delitzsch	5,3	26,2	0,3
Aufnahme: Klinikum Hoyerswerda gGmbH	Entlassung: Krankenhaus Weisswasser gGmbH	4,4	76,6	0,9
Entlassung: Krankenhaus Weisswasser gGmbH	Aufnahme: Klinikum Hoyerswerda gGmbH	4,4	60,8	0,9
Aufnahme: Oberlausitz-Kliniken gGmbH Krankenhaus Bautzen	Entlassung: Oberlausitz-Kliniken gGmbH Krankenhaus Bischofswerda	13,4	55	0,4
Entlassung: Oberlausitz-Kliniken gGmbH Krankenhaus Bischofswerda	Aufnahme: Oberlausitz-Kliniken gGmbH Krankenhaus Bautzen	13,4	27,1	0,4
Entlassung: Paracelsus-Klinik Zwickau	Aufnahme: Heinrich-Braun-Klinikum Zwickau gGmbH	5,1	56,5	0,5
Aufnahme: Heinrich-Braun-Klinikum Zwickau gGmbH	Entlassung: Paracelsus-Klinik Zwickau	5,1	17,1	0,5
Entlassung: Paracelsus-Klinik Reichenbach GmbH	Aufnahme: Vogtlandklinikum Plauen GmbH	5,1	27,5	0,3
Aufnahme: Vogtlandklinikum Plauen GmbH	Entlassung: Paracelsus-Klinik Reichenbach GmbH	5,1	23,7	0,3
Aufnahme: Park-Krankenhaus Leipzig-Südost GmbH	Entlassung: Herzzentrum Leipzig GmbH	3,9	40	0,4
Entlassung: Herzzentrum Leipzig GmbH	Aufnahme: Park-Krankenhaus Leipzig-Südost GmbH	3,9	19,4	0,4

Aufnahme: FACHKRANKENHAUS MARIENSTIFT SCHWARZENBERG	Entlassung: HELIOS Klinikum Aue	4,9	32,4	0,4

Tabelle 5: Assoziationsregeln Verlegungen mit HD

Bedingung	Folgerung	Lift	Konfidenz	Support
Entlassung: Klinikum Mittleres Erzgebirge gGmbH - Haus Zschopau	Aufnahme: Klinikum Chemnitz gGmbH	5,3	79,3	2,2
Aufnahme: Klinikum Chemnitz gGmbH	Entlassung: Klinikum Mittleres Erzgebirge gGmbH - Haus Zschopau	5,3	14,7	2,2
Entlassung: Landkreis Mittweida Krankenhaus gGmbH	Aufnahme: Klinikum Chemnitz gGmbH	4,4	65,3	1,8
Aufnahme: Klinikum Chemnitz gGmbH	Entlassung: Landkreis Mittweida Krankenhaus gGmbH	4,4	12,2	1,8
Entlassung: Klinikum des Landkreises Löbau-Zittau gGmbH	Aufnahme: Städtisches Klinikum Görlitz gGmbH	13,4	60,1	1,7
Aufnahme: Städtisches Klinikum Görlitz gGmbH	Entlassung: Klinikum des Landkreises Löbau-Zittau gGmbH	13,4	37,5	1,7
Entlassung: DRK-Krankenhaus Chemnitz-Rabenstein	Aufnahme: Klinikum Chemnitz gGmbH	5,1	75,5	1,4
Aufnahme: Klinikum Chemnitz gGmbH	Entlassung: DRK-Krankenhaus Chemnitz-Rabenstein	5,1	9,5	1,4
Entlassung: Städtisches Krankenhaus Dresden-Neustadt	Aufnahme: Herzzentrum Dresden GmbH	5,1	62,4	1,3
Aufnahme: Herzzentrum Dresden GmbH	Entlassung: Städtisches Krankenhaus Dresden-Neustadt	5,1	10,6	1,3
Entlassung: Universitätsklinikum Carl Gustav Carus an der TU Dresden, AöR	Aufnahme: Herzzentrum Dresden GmbH	3,9	47,4	1,2
Aufnahme: Herzzentrum Dresden GmbH	Entlassung: Universitätsklinikum Carl Gustav Carus an der TU Dresden, AöR	3,9	9,7	1,2
Entlassung: Diakoniekrankenhaus Chemnitzer Land- DIAKOMED gGmbH	Aufnahme: Klinikum Chemnitz gGmbH	4,9	72,5	1,2
Aufnahme: Klinikum Chemnitz gGmbH	Entlassung: Diakoniekrankenhaus Chemnitzer Land- DIAKOMED gGmbH	4,9	7,9	1,2
Entlassung: Weißeritztal-Kliniken GmbH	Aufnahme: Herzzentrum Dresden GmbH	4,3	51,7	1,2

Tabelle 6: Assoziationsregeln Verlegungen ohne HD

Evaluation

Bei der Analyse der Assoziationsregeln ist aufgefallen, dass viele nah beieinander liegende Krankenhäuser viele Verlegungen untereinander durchführen. Ebenso werden viele Fälle in spezielle Krankenhäuser, wie z. B. die Herzzentren verlegt. Es wurden aber keine großen Verlegungen zwischen Krankenhausketten festgestellt.

4.3.3 Fragestellung 3

Business Understanding

Mit Hilfe der folgenden Aufgabenstellung wird das Business Understanding hauptsächlich dargestellt.

Die InEK-Daten sollen darauf hin untersucht werden, wie sich in den einzelnen Krankenhäusern im Freistaat Sachsen und in ausgewählten Fachgebieten Neurochirurgie, Neurologie, Innere Medizin und Chirurgie das Leistungsspektrum (DRG und/oder OPS) bei Hauptdiagnose C 71 Gehirnkrebs über die Jahre entwickelt hat.

Für diese Fragestellung wurden mehrere Klassifikationen bzgl. der jeweiligen Jahre für die verschiedenen Krankenhäuser als Data Mining Aufgabe untersucht.

Data Understanding

Für diese Fragestellung benötigt man die Faktentabellen FAKT_FAELLE, FAKT_ICD, FAKT_FAB, FAKT_OPS und FAKT_ENTGELTE. Darüber hinaus benötigt man noch die Dimensionstabellen für die DRG Bewertungsreserven.

Bei der „Data Preparation“ müssen die Daten so vorbereitet werden, dass eine Klassifikation die der Fragestellung gerecht wird durchgeführt werden kann. Dabei muss zuerst aus den vorhandenen Daten die zu untersuchende Teilmenge ermittelt werden. Dafür müssen für die zwei Teilaufgaben unterschiedliche Untersuchungsmengen gebildet werden. Für die OPS-Aufgabe wird die View „2_GEHIRNKREBS_CROSS_FAB_OPS“, wie in Abbildung 10 gezeigt, erstellt.

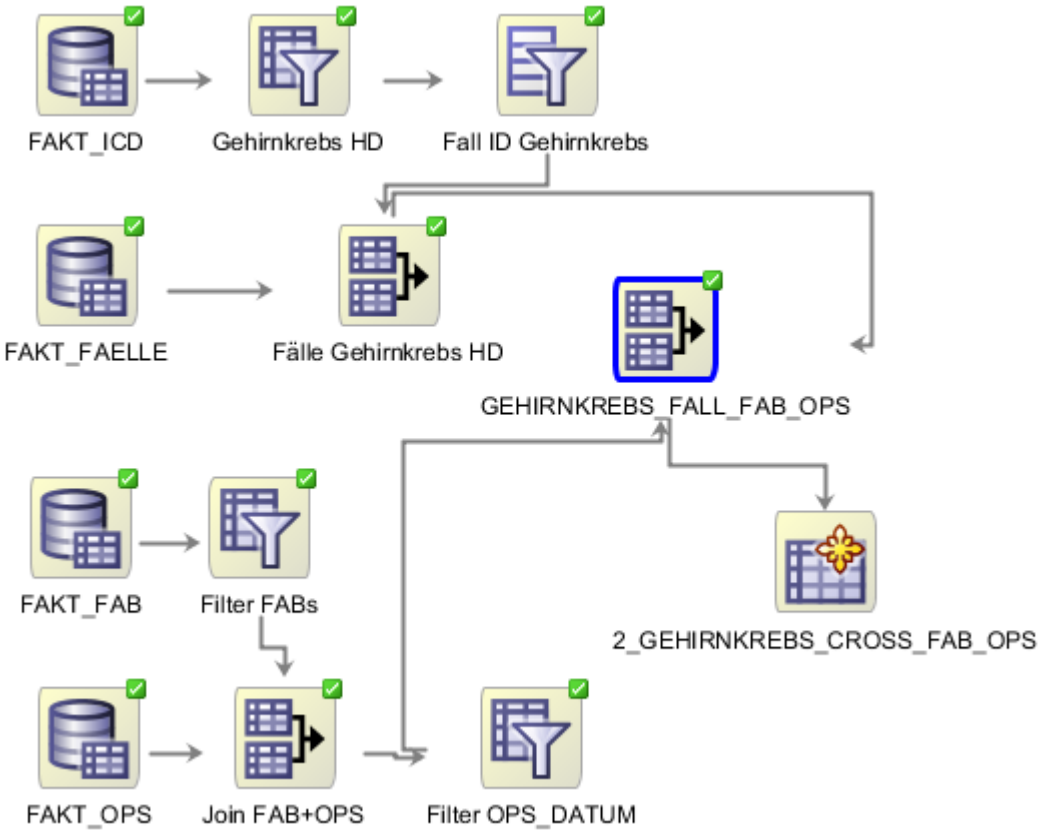

Abbildung 10: Erstellung von 2_GEHIRNKREBS_CROSS_FAB_OPS

Dabei werden alle Fälle mit der Hauptdiagnose Gehirnkrebs in „Gehirnkrebs HD“ mit folgender Filterregel ausgewählt:

```
"DIAGNOSEART" = 'HD' and "ICD_KODE" like 'C71%'
```

In „Fall ID Gehirnkrebs“ werden alle Attribut bis aus FAFALL_ID herausgefiltert, da die restlichen Attribute nicht mehr benötigt werden.

In „Fälle Gehirnkrebs HD“ befinden sich dementsprechend alle Fälle, bei denen als Hauptdiagnose Gehirnkrebs festgestellt wurde. Dabei wird die Tabelle FAKT_FAELLE mit den IDs aus „FALL ID Gehirnkrebs“ verbunden.

Die in der Aufgabenstellung beschriebenen Fachabteilungen werden in „Filter FABs“ aus der Tabelle FAKT_FAB ausgewählt. Dabei wird das Attribut FAB aus FAKT_FAB zur Entscheidung verwendet. Dieses Attribut wird in (Deutsche Krankenhausgesellschaft, 2011) im Dokument „Datenübermittlung nach § 301 Abs. 3 SGB V“ unter „Anlage 2 Schlüsselverzeichnis“, „Schlüssel 6 Fachabteilungen (BPflV)“ beschrieben. Dabei hat die Neurochirurgie das Kennzeichen 1700, die Neurologie das Kennzeichen 2800 und die Innere Medizin das Kennzeichen 0100. Dabei muss man beachten, dass die beiden letzten Stellen sogenannte Schwerpunkte in der Fachabteilung darstellen. Die Regel zur Auswahl lautet demnach wie folgt:

```
("FAB" >= 100 and "FAB" < 200) or ("FAB" >= 1700 and "FAB" < 1800) or ("FAB" >= 2800 and "FAB" < 2900)
```

In „Join FAB+OPS“ werden die beiden Tabellen anhand des Attributes FAFALL_ID miteinander verbunden. Mit „Filter OPS_Datum“ werden schließlich nur die OPS, die in einer Fachabteilung durchgeführt wurden ausgewählt. Dabei kommt die folgende Regel zum Einsatz:

```
"OPS_DATUM" >= "DATUM_AUFNAHME" and "OPS_DATUM" <= "DATUM_ENTLASSUNG"
```

Die beiden erzeugten Tabellen „Filter OPS_DATUM„ und „Fälle Gehirnkrebs HD“ werden anschließend anhand der Attribute ID und FAFALL_ID in GEHIRNKREBS_FALL_FAB_OPS miteinander verbunden.

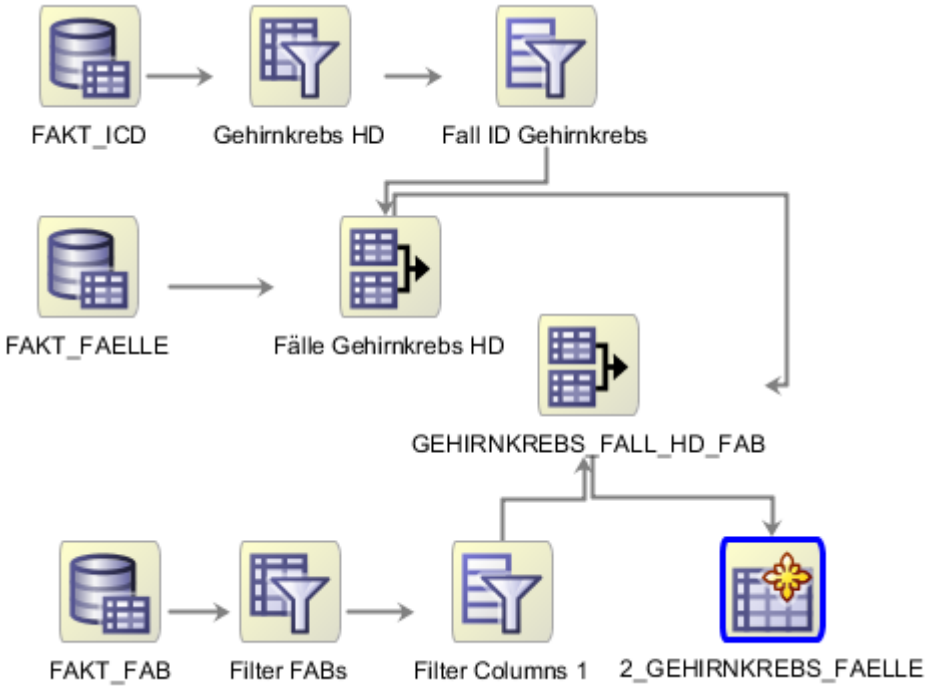

Abbildung 11: Erstellung von 2_GEHIRNKREBS_FAELLE

Für die DRG-Aufgabe muss man die Vorbereitung anders angehen. Da die benötigten DRG-Fallpauschalen in der Tabelle FAKT_ENTGELTE liegen und man diese nicht wie die Tabelle FAKT_OPS mit der Tabelle FAKT_FAB verbinden kann, muss man eine andere Lösung finden. Dabei wurden nur die Fälle berücksichtigt, die irgendwann im Laufe des Aufenthalts in eine der benötigten Fachabteilungen aufgetreten sind.

Data Preparation

Für die „Data Preparation“ der OPS Aufgabe müssen die Daten in eine Form gebracht werden, so dass eine Klassifikation möglich ist. Anhand Abbildung 12 kann man erkennen, dass aus der Tabelle „2_GEHIRNKREBS_CROSS_FAB_OPS“ mit Hilfe des Knotens „Filter IK“ ein Krankenhaus ausgewählt wird. In diesem Fall wurde sich für das Krankenhaus mit dem IK 261400530, Klinikum St. Georg gGmbH, entschieden, da Krankenhäuser mit wenigen Daten pro Jahr untersucht werden sollten. In diesem Falle wurden aber nur Krankenhäuser die in wenigs-

tens einem Jahr mehr als 50 Fälle zur Verfügung haben berücksichtigt. Dadurch kann man untersuchen, ob sich das Leistungsspektrum von einem Jahr zu einem anderen verändert hat oder nicht. An Daten über das Leistungsspektrum für Krankenhäuser mit den meisten Fällen pro Jahr war man von Seiten des SMS nicht interessiert. Für die Analyse musste noch zusätzlich die Dimensionstabellen für OPS mit einbezogen werden, da diese Daten notwendig sind für die Schätzung des Leistungsspektrums.

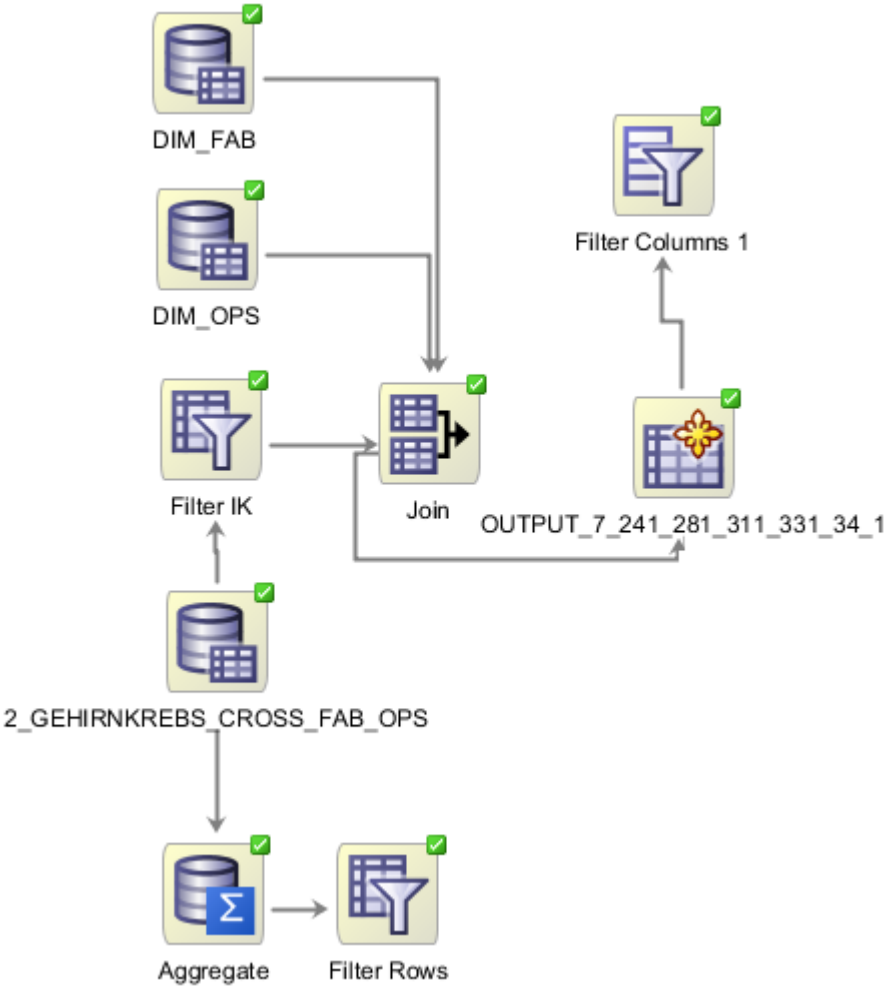

Abbildung 12: Workflow zur Fragestellung 3 – OPS (Datenauswahl)

Es sind die folgenden Transformationen nötig um die Daten klassifizieren zu können.

OPS_KODE wurde folgendermaßen in ein neues Attribut OPS_KODE_ transformiert:

```
substr(replace(ops_kode, '-', ''), 1, 4)
```

Das ersetzen von '-' mit '' ist nötig, da für das Attribut OPS_KODE im Zeitraum von 2003-2009 die Codierung geändert wurde. Es werden nur die ersten 4 Stellen von OPS_KODE betrachtet, da ansonsten zu viele verschiedene Werte für OPS_KODE auftreten, bei der die Klassifikation zu wenig Trennschärfe aufzeigen würde. Das Attribut Maßnahme kann man nicht für eine Klassifikation benutzen, da sich die Bezeichnungen teil-weise verändern.

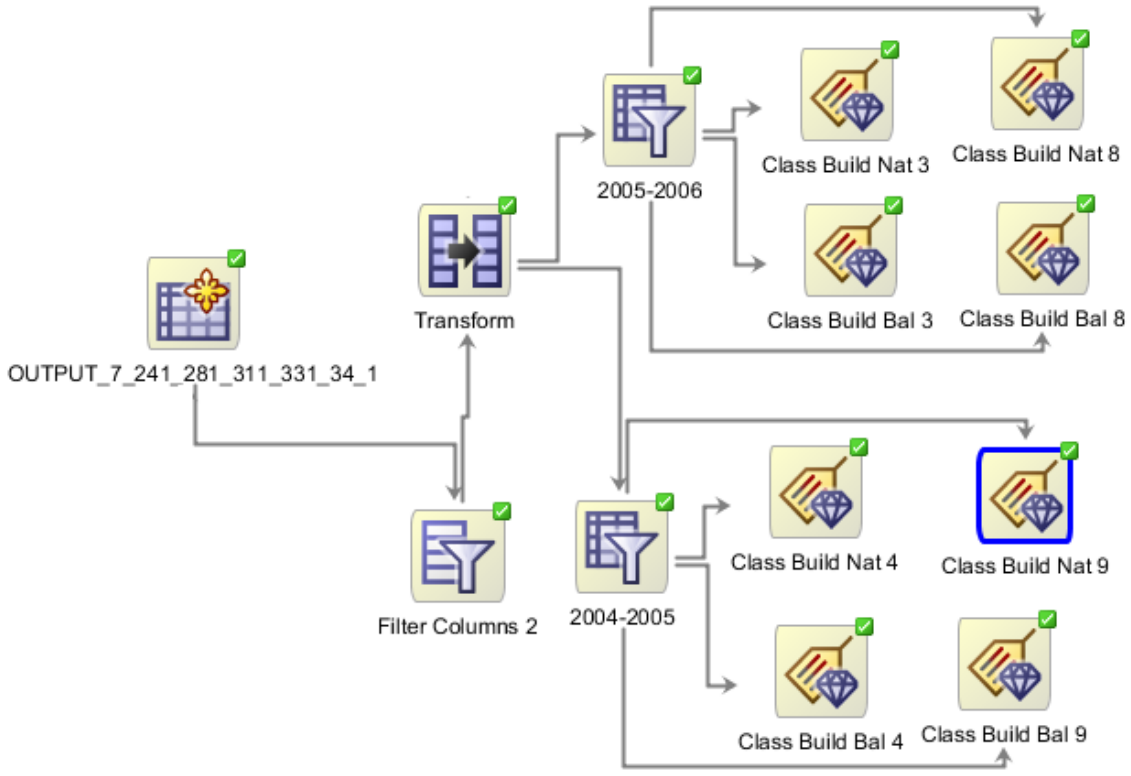

Abbildung 13: Workflow zur Fragestellung 3 – OPS (Auszug aus der Klassifikation)

Bei der DRG-Klassifikation müssen die zu den Fällen aus „2_GEHIRNKREBS_FAELLE“ gehörigen DRG aus der Faktentabelle

FAKT_ENTGELTE eingebunden werden. Dabei wird die Dimensionstabellen DIM_DRG_BWR_HA benötigt um die benötigten Attribute für die DRG-Fallpauschalen auszuwählen. Die Verbindung wird in „Join 1“ mittels des Attributs DRG_SCHLUESSEL durchgeführt. In „Filter DRG Jahre“ werden alle DRG herausgefiltert, die nicht im Jahr der Datenerhebung des jeweiligen Falles aufgeführt sind. Das geschieht mit der Filterregel:

```
"JAHR" = "JAHR_DATENERHEBUNG"
```

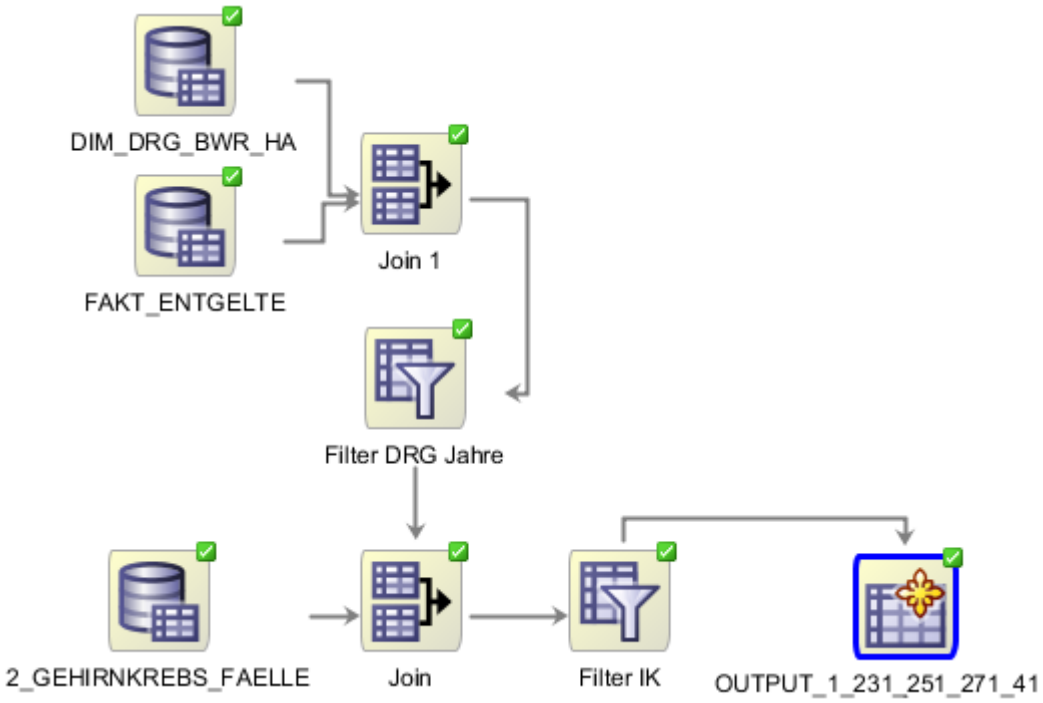

Abbildung 14: Workflow zur Fragestellung 3 – DRG (Datenauswahl)

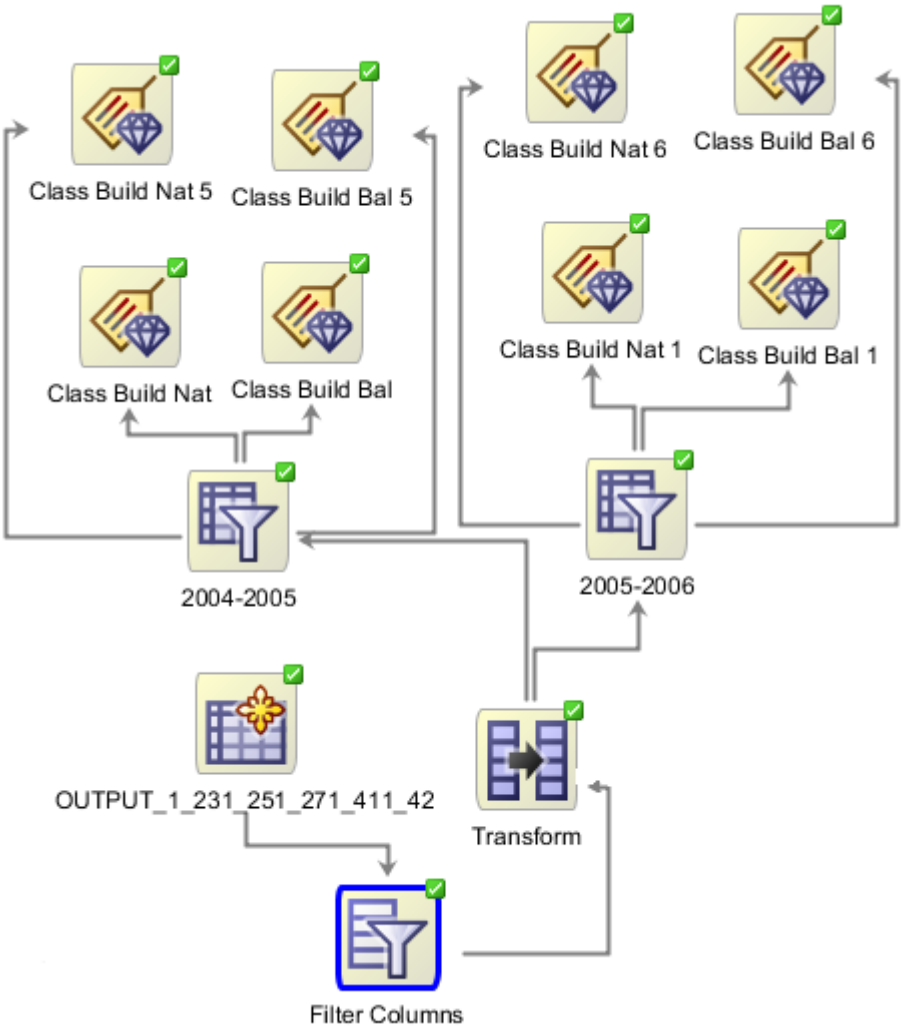

Abbildung 15: Workflow zur Fragestellung 3 – DRG (Auszug aus der Klassifikation)

Für die DRG-Klassifikation wurden nur die Attribute berücksichtigt, die für eine Identifizierung eines DRG-Spektrums nötig sind. Dafür wurden u. a. folgende Attribute ausgewählt: Bezeichnung, DRG_Schluessel und DRG_BEZ. Dabei mussten noch einige Transformationen durchgeführt werden. Für DRG_Schluessel wurde ein neues Attribut DRG_Schluessel_substr folgendermaßen eingeführt, das nur die ersten beiden Stellen des DRG_Schluessels beinhaltet. Dadurch kann man die Anzahl dieses Attributs ein bisschen einschränken. Bei Bezeichnung muss man beachten, dass diese von Jahr zu Jahr verschieden sind. Dabei

bleiben aber die ersten Wörter meist gleich, so dass man mittels der Transformation der Variable in BEZEICHNUNG_SUBSTR_1_30 durch

```
SUBSTR("BEZEICHNUNG", 1, 30)
```

einen besseren Klassifikator erhält. Dadurch werden ähnliche Bezeichnungen eliminiert.

Modeling

Nachdem die Daten geeignet transformiert wurden muss eine Klassifikation angewendet werden. Dabei werden die Daten anhand des Jahres der Datenerhebung aufgeteilt und eine Klassifikation anhand des Jahres der Datenerhebung durchgeführt. Es wurden für jedes Jahr 4 Klassifikationen durchgeführt.

Bei Class Build Bal – Class Build Bal 4 wurde bei Performance Settings unter den Advanced Settings für die Klassifikation Balanced ausgewählt. Dadurch werden Klassen die weniger oft auftreten stärker gewichtet, so dass diese Klassen nicht wegen der ungünstigen Klassenverteilung stärker gewichtet werden.

Bei Class Build Nat – Class Build Nat 4 wurde bei Performance Settings unter den Advanced Settings für die Klassifikation Natural ausgewählt. Dadurch wird die natürliche Klassenverteilung, wie sie in den Ursprungsdaten vorhanden ist, für die Klassifikation verwendet. Dadurch werden die Klassen nicht stärker gewichtet, falls sie weniger oft auftreten.

Des Weiteren wurde aufgrund bei Class Build Bal/Nat 5 - Class Build Bal/Nat 9 die Klassifikationsgenauigkeitsmessung anhand der kompletten

Datenbasis vorgenommen und nicht wie üblich nur an einem Teilbereich von 60%.

Bei der Analyse der Daten wurden u. a. die in Abbildung 16 dargestellte Klassifikationsregel gefunden. Für die anderen Jahre sieht das ähnlich aus.

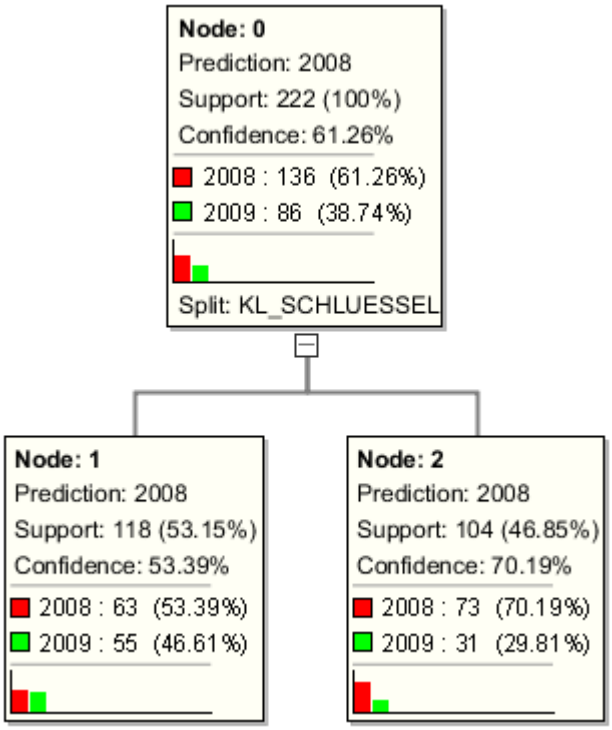

Abbildung 16: Klassifikationsregel für OPS für 2008-2009

	2004-2005	2005-2006	2006-2007	2007-2008	2008-2009
SVM	28,4645	36,7417	36,76	26,9084	48,0677
Naive Bayes	0	0	0	0	0
Verallg. lin. Modelle	25,7165	0	0	27,0207	20,554
Entscheidungsbaum	10,3165	19,3844	7,2604	8,6383	17,63

Tabelle 7: Testergebnisse für das Klinikum St. Georg gGmbH OPS balanced

In Tabelle 7 werden die Testergebnisse für eine OPS-Klassifikation dargestellt. Dabei handelt es sich um eine mit den Performance Settings balanced trainierte und mit allen Trainingsdaten getestete Klassifikation.

Bei der DRG-Klassifikation wird die Klassifikation analog zur OPS-Klassifikation durchgeführt. Dabei wurde u. a. die Paracelsus-Klinik in Zwickau untersucht. Eine Regel für 2005-2006 kann man in Abbildung 17 anschauen. Eine Beispielsregel, die man aus dem Entscheidungsbaum bekommen kann ist

falls(BEZEICHNUNG_SUBSTR_1_30 isIn ("Beatmung > 999 und 1800 Stun" "Komplexe Kraniotomie oder Wirb" "Kraniotomie oder große Wirbels" "Mäßig komplexe Kraniotomie")) dann ist die Klasse 2006

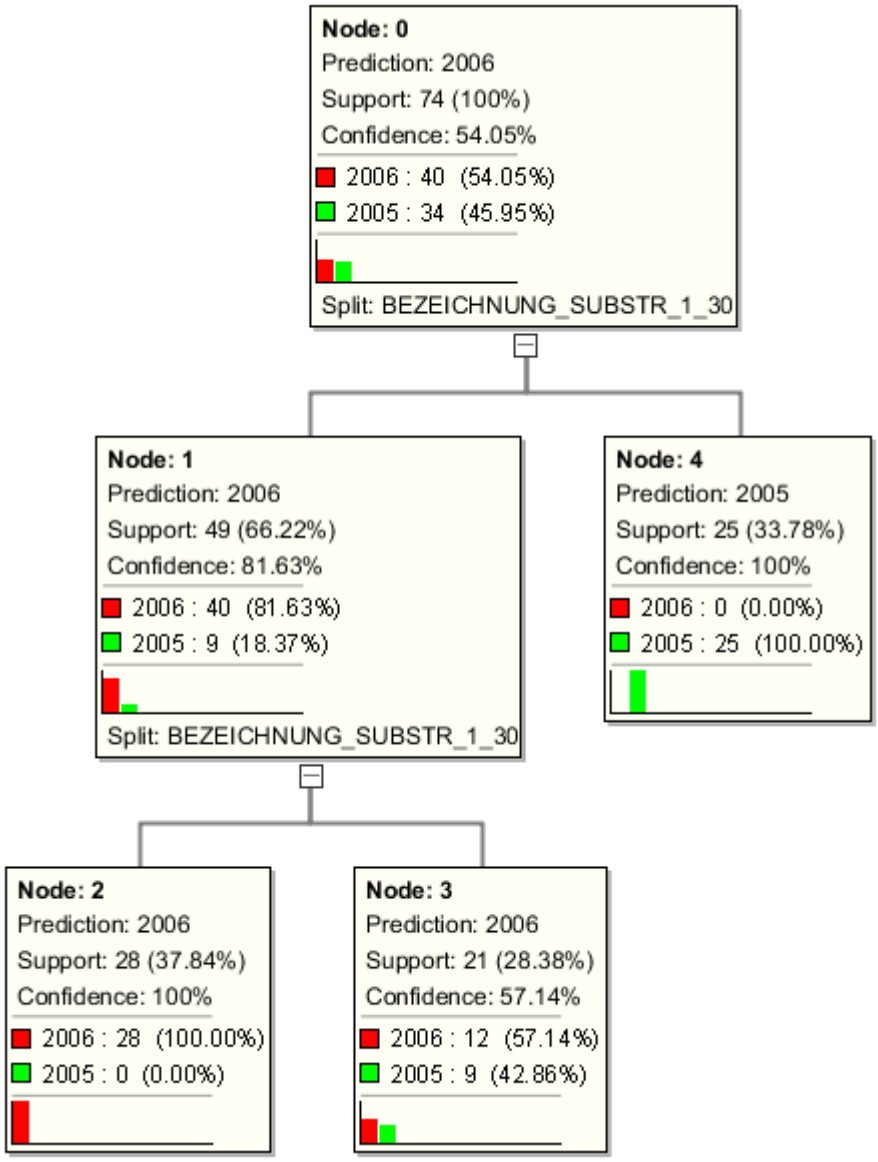

Abbildung 17: Klassifikationsregel für DRG für 2005-2006

Evaluation

Bei der Analyse der gewonnenen Regeln fällt auf, dass die Klassifikationsregel immer vom Attribut KL_SCHLUESSEL gebildet wird. Das liegt daran, dass dieses Attribut numerisch ist. Dadurch sinkt die Anwendbarkeit des Entscheidungsbaumes für diese Fragestellung. Nichtsdestotrotz ist es oftmals hilfreich zu erfahren, falls zwischen 2 Jahren ein Wechsel zwischen den Mehrheitsklassen stattfindet. Da der Entscheidungsbaum aufgrund der Einfachheit des Algorithmus aber nicht alle Entscheidungsgrenzen finden kann ist es oftmals notwendig weitere Testergebnisse von Algorithmen darzustellen. Dabei kann man sehen, dass die Support Vector Machine und die GLM (verallgemeinerten linearen Modelle) im Schnitt besser abschneiden als die anderen verfügbaren Algorithmen. Es ist nicht ausreichend gelungen die Datenbasis mittels eines Entscheidungsbaums aufzuteilen. Das Problem bei der Support Vector Machine und den GLM ist, das diese Algorithmen keine verständlichen Regeln liefern. Für die Klassifikationen sollte man immer die Fehlerrate des Klassifikators abschätzen.

Bei der Analyse der DRG-Regeln sieht man, dass für viele Jahre Klassifikationsregeln existieren. Dabei sind aber bei genauer Betrachtung teilweise die Klassifikationsregeln für gleiche Jahre bei verschiedenen Krankenhäusern ähnlich zueinander. Daraus lässt sich schließen das die Bezeichnung beim DRG-Leistungsspektrum evtl. nur in bestimmten Jahren großflächig geändert wird und dadurch eine Unterscheidung bei der Klassifikation zwischen den Jahren entsteht.

5 Fazit

In dieser Arbeit wurden die grundlegenden Ideen und Problemtypen des Data Mining vorgestellt. Einige Algorithmen zu den Problemtypen wurden dargelegt. Es wurde ein Ablaufplan für das Data Mining betrachtet. Daraufhin wurde der Oracle Data Miner dargestellt um damit Fragestellungen vom SMS, mit Hilfe der Anwendung von Data Mining Methoden auf die InEK-Daten, bearbeiten zu können.

Während der Analyse der InEK-Daten mit Data Mining Algorithmen ist vor allem klar geworden, dass man ohne ein Ziel oder eine Idee nach was man in den Daten suchen könnte keine sinnvolle Analyse durchführen kann. Man kann zwar Muster finden, kann sie aber nicht beurteilen, da das grundlegende Verständnis für die Daten nicht notwendigerweise gegeben ist. Durch die Anwendung und Befolgung des CRISP-DM konnte eine sinnvolle Analyse durchgeführt werden.

Die Teststudie konnte einige erstaunliche Ergebnisse zu Tage fördern. Als besonders flexibel hat sich dabei die Assoziationsanalyse erwiesen. Dabei musste man allerdings beachten, dass die Kennzahlen Support, Konfidenz und Lift teilweise anders interpretiert werden mussten.

Nach der Analyse einiger Daten mit dem Oracle Data Miner ist klar geworden, dass der Oracle Data Miner hauptsächlich für kunden- oder marketingorientiertes Data Mining ausgelegt ist. Das kann man bei der Assoziationsanalyse erkennen. Man muss, falls man keine Warenkorbanalyse durchführen möchte mit Hilfe von SQL Befehlen die Analyse vorbereiten. Bei der Warenkorbanalyse werden vorrangig transaktionalen

Daten untersucht. Dadurch ist es schwierig den Oracle Data Miner zu benutzen, wenn man keine oder nur wenige SQL-Kenntnisse hat.

Bei den Algorithmen zur Clusteranalyse ist es momentan nicht möglich sich eine grafische Repräsentation der Cluster anzuschauen. Des Weiteren sind keine Möglichkeiten gegeben mittels eines Verbesserungsverfahrens wie Boosting oder Bagging die Klassifikationsleistung zu erhöhen. Beides könnte man jedoch über eines der vorhandenen Programmierinterfaces nachimplementieren.

Bei der Bearbeitung der Fragestellungen wurde weiterhin klar, dass manche Fragestellungen nicht für das Data Mining geeignet sind. Das kann man bei der ersten Fragestellung erkennen. Es ist dabei oftmals besser die Daten mit OLAP, SQL oder statistischen Auswertungenzu untersuchen. Beim Data Mining werden hauptsächlich Zusammenhänge gesucht und gefunden, die mit diesen anderen Tools entweder gar nicht oder nur bei eher kleinen Datenmengen gefunden werden können. Beim Data Mining ist es weiterhin oftmals nicht gefragt wo sich etwas verändert hat, sondern warum.

Bei der zweiten Fragestellung hat sich gezeigt, dass man mit Data Mining Methoden sehr gut viele Fälle miteinander vergleichen kann um Auffälligkeiten festzustellen. Diese Fälle kann man danach näher untersuchen um evtl. zu erkennen warum diese Fälle diese Auffälligkeiten besitzen.

Für die dritte Fragestellung hat sich gezeigt, dass die Klassifikation ein probates Mittel darstellt um die prinzipiellen Unterschiede zwischen zwei Datenmengen mit einer gegenseitigen Klassifizierung zu erkennen. Dabei muss man allerdings beachten, dass keine sinnvolle Klassifikation

vorgenommen werden kann, wenn es keinen großen Unterschied zwischen den beiden Klassen gibt. Es ist aus diesem Grunde ratsam bei der dritten Fragestellung alle Krankenhäuser zu untersuchen, bis man einen Überblick über das DRG / OPS Leistungsspektrum erhält. Damit kann man dann besonders gut erkennen ob ein Klassifikator aufgrund der Veränderung eines Namens von einer Maßnahme / Bezeichnung oder aufgrund von Veränderungen am Leistungsspektrum des Krankenhauses Unterschiede feststellt.

Literaturverzeichnis

Alpar, P., & Niedereichholz, J. (2000). *Data Mining im praktischen Einsatz*. Braunschweig/Wiesbaden: Vieweg.

Alpaydin, E. (2004). *Introduction to Machine Learning*. Cambridge: MIT Press.

Bamberg, D. G., & Bauer, D. habil F. (1998). *Statistik*. München: Oldenbourg.

Burges, C. J. C. (1997). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.122.3829&rep=rep1&type=pdf (abgerufen am 14.03.2011)

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0 Step-by-step data mining guide. http://www.crisp-dm.org/CRISPWP-0800.pdf (abgerufen am 17.03.2011)

Deutsche Krankenhausgesellschaft. (2011). Datenübermittlung nach § 301 Abs. 3 SGB V. http://www.dkgev.de/media/file/8892.v301_2010-12-17.pdf (abgerufen am 15.02.2011)

Ester, M., & Sander, J. (2000). *Knowledge Discovery in Databases: Techniken und Anwendungen*. Berlin: Springer.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (1996). *Advances in Knowledge Discovery and Data Mining*. Cambridge: MIT Press.

Han, J., & Kamber, M. (2001). *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning*. New York: Springer.

Mitchell, T. M. (1997). *Machine Learning*. Boston: McGraw-Hill.

Netezza Corporation. (2004). Business Intelligence in a Real-Time World. http://download.101com.com/pub/tdwi/files/BI_in_a_Real-Time_World_Sept04.pdf (abgerufen am: 11.05.2011)

Oracle Corporation. (2010). *Oracle Data Mining Concepts* (Vol. 2).

Robotron Datenbank-Software GmbH. (2010a). SMS-ZFD Betriebshandbuch.

Robotron Datenbank-Software GmbH. (2010b). *SMS-ZFD Dokumentation*.

Witten, I. H., & Frank, E. (2001). *Data Mining, Praktische Wekrzeuge und Techniken für das maschinelle Lernen*. München: Hanser.

Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 01.08.2011

Jens Böttcher